Linkage Identification by Nonlinearity Check for Real-Coded Genetic Algorithms

Masaru Tezuka[1,2], Masaharu Munetomo[1], and Kiyoshi Akama[1]

[1] Hokkaido University,
Kita 11, Nishi 5, Kita-ku, Sapporo, 060-0811, JAPAN
tezuka@uva.cims.hokudai.ac.jp
{munetomo, akama}@cims.hokudai.ac.jp
[2] Hitachi East Japan Solutions, Ltd.
2-16-10, Honcho, Aoba-ku, Sendai, 980-0014, JAPAN
tezuka@hitachi-to.co.jp

Abstract. Linkage identification is a technique to recognize decomposable or quasi-decomposable sub-problems. Accurate linkage identification improves GA's search capability. We introduce a new linkage identification method for Real-Coded GAs called LINC-R (Linkage Identification by Nonlinearity Check for Real-Coded GAs). It tests nonlinearity by random perturbations on each locus in a real value domain. For the problem on which the proportion of nonlinear region in the domain is smaller, more perturbations are required to ensure LINC-R to detect nonlinearity successfully. If the proportion is known, the population size which ensures a certain success rate of LINC-R can be calculated. Computational experiments on benchmark problems showed that the GA with LINC-R outperforms conventional Real-Coded GAs and those with linkage identification by a correlation model.

1 Introduction

The optimization by means of Genetic Algorithms (GAs) is promoted by the exchange of building blocks (BBs). BBs are considered sub-solutions and important subcomponents. GA's search capability is improved by identifying BBs accurately and preventing crossover operators from destructing BBs. Thus, linkage identification, the procedure to recognizing BBs, plays an important role in GA's optimization. Many research efforts have been concentrated on the linkage identification of binary-coded GAs. LLGA (Linkage Learning GA) [1] applies a two-point-like crossover operator to ring-shaped chromosomes and construct linkages dynamically. BOA (Bayesian Optimization Algorithm)[2] constructs a Bayesian network based on the distribution of individuals in a population and identifies linkages indirectly. LINC (Linkage Identification by Nonlinearity Check) [3], LIMD (Linkage Identification by non-Monotonicity Detection) [4], and LIEM (Linkage Identification with Epistasis Measure) [5] identifies linkage groups based on the nonlinearity, non-monotonicity, and epistasis measures between loci respectively.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3103, pp. 222–233, 2004.

Real-Coded GAs operate on real-valued vectors as their chromosomes [6] and have been applied to various real value optimization problems[7,8,9]. When an objective function in the domain of the reals is decomposable or quasi-decomposable to a number of lower order sub-functions, the solution of the original problem is yielded by combining independently obtained sub-solutions. Solving the sub-problems is considered as obtaining BB candidates in the domain of the reals. Thus, accurate linkage identification methods for Real-Coded GAs are desired.

While many researches on the linkage identification for binary-coded GAs have been conducted, only a few are known for Real-Coded GAs. Piecewise Interval Correlation by Iteration (PICI) [10,11] is one of them. PICI calculates correlation coefficients among loci, and recognizes the set of loci with high correlations as a linkage group. However, high correlation means that there is *linear* relation among the loci. The loci with linear relation are still additively decomposable into much smaller sub-problems. Each linkage group as a set of unseparable portions should have *nonlinear* relation. Thus, we introduced a new linkage identification method for Real-Coded GAs called LINC-R (Linkage Identification by Nonlinearity Check for Real-Coded GAs), based on the nonlinearity among loci.

In section 2, we briefly review PICI. Then we propose a new method, LINC-R in section 3. LINC-R is compared to PICI and a conventional Real-Coded GA. The results of the comparison are reported in section 4, followed by conclusion in section 5.

2 Piecewise Interval Correlation by Iteration

The probability density function of individual $\mathbf{x}$ at generation t under proportionate selection is written as

$$p(\mathbf{x}, t) = \frac{f^t(\mathbf{x}) p_0(\mathbf{x})}{\int f^t(\mathbf{x}) p_0(\mathbf{x}) d\mathbf{x}} \tag{1}$$

where f is a non-negative fitness function and p_0 is an initial distribution of individuals. The distribution of individuals in a population reflects the landscape of f and that gets amplified as t increases. If there are linkages among the arguments of f, there also may be some degree of correlation among them. PICI divides a domain into several sub-domains and calculate correlation coefficients of the sub-domains. Then the weighted average of the coefficients is taken as the piecewise interval correlation. PICI recognizes the set of loci with high correlations as a linkage group.

PICI has two variations, Linkage Identification with Single-Stage evolution (LISS) and with Multi-Stage evolution (LIMS). LISS learns linkages as optimization progresses. LIMS consists of three stages, initial, learning and searching stage. Firstly, in the initial stage, population forms a linkage structure. Then in the learning stage, LIMS learns linkages. Finally in the searching stage, LIMS uses the linkage information in optimization. It is reported that LIMS has higher performance than LISS.

3 GA with Linkage Identification by Nonlinearity Check for Real-Coded GAs

3.1 Nonlinearity Check for Real-Coded GAs

In this section, we propose Linkage Identification by Nonlinearity Check for Real-Coded GAs (LINC-R). LINC-R is based on an idea that if a function is linearly decomposable to several sub-functions and two loci belong to different sub-functions, the partial difference of the function with respect to one of the loci is independent from the value on the other locus. LINC-R is also an extension of LINC which is proposed for binary GAs since both LINC and LINC-R identify linkages based on nonlinearity detection.

LINC checks nonlinearity in each pair of loci whether $\Delta f_i + \Delta f_j = \Delta f_{ij}$ or not, where Δf_i is the amount of change caused in fitness by a perturbation on locus i, Δf_j is that caused by a perturbation on locus j, and Δf_{ij} is that by the perturbations on both i and j at a time. In other words, LINC judges the loci have linkage when the following condition is satisfied.

$$|\Delta f_{ij} - (\Delta f_i + \Delta f_j)| > \epsilon \tag{2}$$

where ϵ is a parameter specifying allowable error for nonlinearity check.

Since the chromosomes of Real-Coded GAs are real-valued vectors, LINC-R can not perturb the value on each locus in the same way that LINC does on binary strings. Thus we introduced random perturbation. LINC-R checks nonlinearity by equation (2) with random perturbation,

$$\Delta f_{ij} = f(x_i + \Delta x_i, x_j + \Delta x_j) - f(x_i, x_j) \tag{3}$$

$$\Delta f_i = f(x_i + \Delta x_i, x_j) - f(x_i, x_j) \tag{4}$$

$$\Delta f_j = f(x_i, x_j + \Delta x_j) - f(x_i, x_j) \tag{5}$$

where Δx_i and Δx_j are chosen randomly so that the perturbed points will fall into the domain.

LINC-R evaluates the objective function values at four points, (x_i, x_j), $(x_i + \Delta x_i, x_j)$, $(x_i, x_j + \Delta x_j)$, and $(x_i + \Delta x_i, x_j + \Delta x_j)$. There is a possibility of misjudgment that in spite of a pair of loci having a linkage the randomly selected four points seem to be linear unexpectedly. In order to reduce the possibility of such misjudgment, a number of individuals are used to identify a linkage. Figure 1 shows the procedure of LINC-R. The number of objective function evaluations required for LINC-R is $\{3n(n-1)/2+1\}|P|$, where $|P|$ is the size of a population P.

The essential difference between LINC-R and LINC is their domain, the binaries or the reals. LINC-R tests nonlinearity by random perturbations in its domain while so does LINC by bit-wise perturbations. In the domain of the binaries, only '0' and '1' can be taken on the locus. However, one random perturbation covers only small region in the domain of the reals. Thus, in order to realize accurate linkage identification, we need to employ properly sized population according to the difficulty of problems.

```
1.  Randomly initialize a population P.
2.  for each x in P
3.    for i = 1 to n-1
4.      for j = i +1 to n
5.        if linkage between locus i and j has not detected yet
6.          dx_i = (Uniform random value [x_i^L, x_i^U]) - x_i.
7.          dx_j = (Uniform random value [x_j^L, x_j^U]) - x_j.
8.          Δf_ij = f(..., x_i + dx_i, ..., x_j + dx_j, ...) - f(..., x_i, ..., x_j, ...)
9.          Δf_i = f(..., x_i + dx_i, ..., x_j, ...) - f(..., x_i, ..., x_j, ...)
10.         Δf_j = f(..., x_i, ..., x_j + dx_j, ...) - f(..., x_i, ..., x_j, ...)
11.         if | Δf_ij - (Δf_i + Δf_j) | > ε then
12.             Detect linkage between locus i and j.
13.         end if
14.       end if
15.     end for
16.   end for
17. end for
```

Fig. 1. LINC-R: Linkage Identification for Real-Coded GA based on nonlinearity

3.2 Population Sizing of the LINC-R

In this section, we will discuss on the population size required for LINC-R. In order to investigate the population size, we introduced a function f_D which has partial nonlinearity in the domain $x_i \in [0.0, 1.0]$ for $i = 1, ..., n$.

$$f_D(\mathbf{x}) = \begin{cases} f_L(\mathbf{x}) + f_{NL}(\mathbf{x}), & if \sum_{i=1}^{n} x_i^2 \leq r^2 \\ f_L(\mathbf{x}), & otherwise. \end{cases} \tag{6}$$

$$f_{NL}(x_1, ..., x_n) = \left(1 - \frac{1}{r}\sqrt{\sum_{i=1}^{n} x_i^2}\right) \tag{7}$$

$$f_L(x_1, ..., x_n) = \frac{\lambda}{n}\sum_{i=1}^{n} x_i \tag{8}$$

where $0.0 \leq \lambda < 1.0$ and $0.0 \leq r \leq 1.0$ are the parameters. The optimal value of f_D is 1.0 at $\mathbf{x} = (0, ..., 0)$ and a sub-optimum is λ at $\mathbf{x} = (1, ..., 1)$. f_D is nonlinear within an r radius from the origin and linear otherwise. Figure 2 shows an example of f_D with $n = 2$, $\lambda = 0.8$, $r = 0.2$.

Since LINC-R uses four points (x_i, x_j), $(x_i + \Delta x_i, x_j)$, $(x_i, x_j + \Delta x_j)$, and $(x_i + \Delta x_i, x_j + \Delta x_j)$ to test nonlinearity, when all the points are outside of the nonlinear region of the objective functions, LINC-R fails to detect linkage. However, it can succeed in detecting linkage as long as at least one of the points is in the nonlinear region. Here, a denotes the proportion of the nonlinear region in the domain. $a = \pi r^2/4$ for f_D. The probability that at least one of four points falls into the nonlinear region is $1-(1-a)^4$. Thus, the probability of successfully

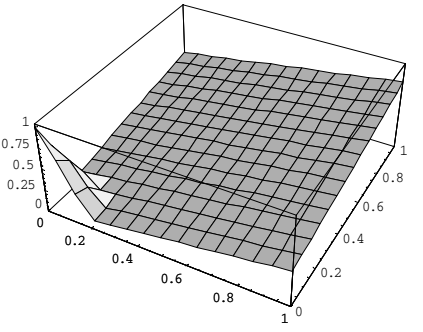

Fig. 2. Objective function f_D which has nonlinerity in a part of domain

identifying linkage in a pair of loci is,

$$Pr = \left\{1 - (1-a)^4\right\} \sum_{i=1}^{|P|} \left\{(1-a)^4\right\}^{i-1} = \left\{1 - (1-a)^4\right\}^{|P|}. \tag{9}$$

The third term is derived from the fact that the second term is the sum of the geometric sequence. From the equation, we have the population size as follows:

$$|P| = \frac{ln(1-Pr)}{4ln(1-a)} \tag{10}$$

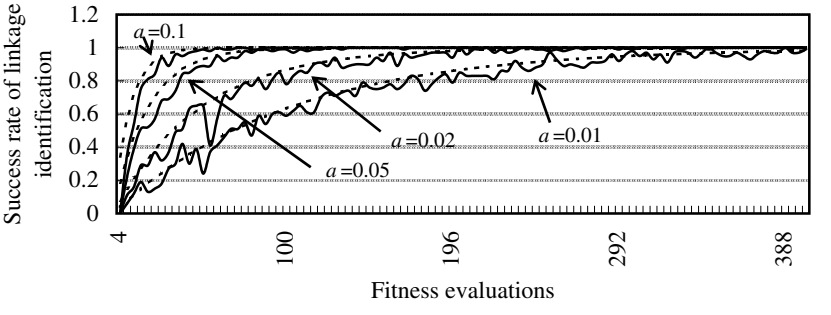

Fig. 3. Success rate of linkage identification along with the proportion of nonlinear region.

Figure 3 shows the success rate of linkage identification with $a = 0.1$, 0.05, 0.02, 0.01. X axis shows the number of objective function evaluations to identify linkage, that is $|P| \times 4$. Dotted lines show theoretical probability obtained from equation (9). Solid lines show the result of computational experiments on f_D where $n = 2$. The values are the average of 100 trial runs. A linkage may be

successfully identified within 100 evaluations if the proportion of the nonlinear region is more than five percent ($a \geq 0.05$). Note that the above discussion concerns the probability of linkage identification for one pair of loci while there are ${}_nC_2$ combinations of possible pairs for n dimensional vectors.

3.3 Optimization Procedure of the GA with Linkage Identification by Nonlinearity Check for Real-Coded GAs

The optimization procedure of real-coded GAs with LINC-R consists of two stages as shown in figure 4. Firstly, in the linkage identification stage, LINC-R identifies linkage groups. Then, in the optimization stage, GAs are performed in each linkage group separately.

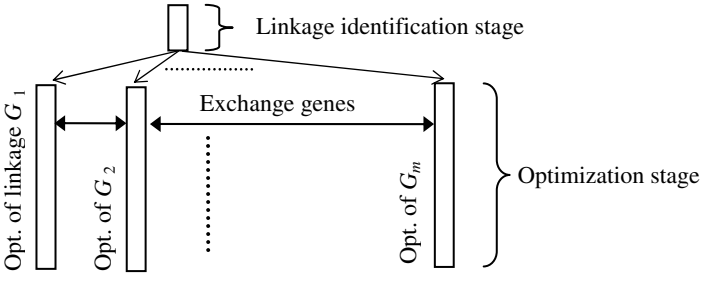

Fig. 4. Optimization by Real-Coded GA with LINC-R.

Here, m denotes the number of linkage groups identified by LINC-R. $G_i = \{x_{k_i-1}, ..., x_{k_i}\}$ is the set of loci belonging to linkage group i. In the optimization stage, m islands are created. The population created in the linkage identification stage is not used in this stage. The population of island i is initialized so that the genes on the loci belonging to G_i are initialized randomly, and for the genes on the other loci, the gene on the same locus has the same value throughout the population. In island i, genetic operators are applied to the only loci in G_i. Thus, island i optimizes linkage group G_i, that is $|G_i|$ dimensional sub-problem. In this stage, genes are exchanged among islands periodically. The genes on the loci in G_i of the best individual of the island i are copied to those of the other islands. Every time genes are exchanged, all the population in each island has to be re-evaluated and the number of function evaluations is consumed largely for the re-evaluation.

Generally, the computational complexity of the optimization problems rises exponentially as the dimension increases. Thus, in the optimization stage, more search effort has to be invested to the island which optimizes larger linkage group. In this paper, we allocate search cost as follows:

The number of generations. Allocate the number of generations proportional to the dimension of linkage group. $|G_i|$ generations are alternated in island i while $|G_j|$ generations are in island j.

Population size. Population sizes are set to be proportional to the square of the dimension. The population size of island i is $|P_i| = C_p \times |G_i|^2$ where C_p is a constant.

4 Numerical Experiments

4.1 Experimental Conditions

We employ SPX (simplex crossover)[12] and MGG (Minimum Generation Gap) model [13] both employed in the reports on PICI [10,11] in order to compare the performance of LINC-R and PICI.

We judge the optimal solution $(o_1, ..., o_n)$ is found when $\forall i,\ x_i \in [o_i - \Delta x/2, o_i + \Delta x/2]$ where Δx is a resolution of the solution. We set Δx to 0.001, same as the PICI's reports used.

We set the interval of gene exchange depending on the total population which is the sum of the population of all the islands. When the total population is smaller than 5,000, genes are exchanged every 50,000 function evaluations. When the total population is smaller than 10,000, genes are exchanged every 100,000 function evaluations, otherwise every 1,000,000 evaluations.

4.2 Functions with Nonlinearity in the Whole Search Space

Firstly, we use Type I and Type II functions employed by Tsutsui et al. in PICI's reports defined by following equations respectively.

$$F_1(\mathbf{x}) = Fr_T(x_1, ..., x_T) + Fs_L(x_{T+1}, ..., x_{T+L}) \tag{11}$$

$$F_2(\mathbf{x}) = \sum_{i=1}^{T} Fr_2(x_{2i-1}, x_{2i}) + Fs_L(x_{2T+1}, ..., x_{2T+L}) \tag{12}$$

where $-2.048 \leq x_i < 2.047$ for all i. These functions are to be minimized and have global optimum of 0.0 at $\mathbf{x} = (1, ..., 1)$.

Fr_T is a T dimensional Rosenbrock function defined by equation (13) and has linkage between the first and the other arguments. Since nonlinearity of this linkage group exists throughout the domain, proportion of the nonlinear region is 1.0, that is, $a = 1.0$.

$$Fr_T(x_1, ..., x_T) = \sum_{i=2}^{T} \left(100\left(x_1 - x_i^2\right)^2 + (x_i - 1)^2\right) \tag{13}$$

Fs_L is an L dimensional sphere function defined by equation (14) that has no linkage.

$$Fs_L(x_1, ..., x_L) = \sum_{i=1}^{L} (x_i - 1)^2 \tag{14}$$

Both type I and II function are nonlinear among the loci in the same linkage group in the whole domain. That means $a = 1.0$ in equation (9) and the

probability of successfully identifying linkage, Pr, is always 1.0. Thus, we set the population size in the linkage identification stage to one.

We test the capability of the GA employing LINC-R with various T, the dimension of Rosenbrock function. $T = 2, ..., 8$ for Type I function and $T = 2, ..., 4$ for Type II function. L, the dimension of the sphere function, is fixed to 20. C_p is set to 10.

Ten runs are performed for each parameter setting. Each run continues until the optimal solution is found or the number of function evaluations reach to 1.0×10^6. The number includes the evaluations for linkage identification.

Table 1 and 2 show the number of successful runs which found the optimum (#Opt) and the mean (MNE) and the standard deviation (STDEV) of the number of function evaluations to find the optimum in the runs which found the optimum. The standard deviations are zero or very small in easy cases because the optimum solutions are found at the first gene exchange in all the trials in the cases.

Table 1. Result of optimization of type I function.

	without linkage identification		PICI(LIMS)		LINC-R		
T	#Opt	MNE	#Opt	MNE	#Opt	MNE	STDEV
2	10	200,033	10	172,420	10	51,074	0
3	3	452,392	10	204,837	10	52,272	0
4	1	441,632	10	222,771	10	67,053	9,703
5	0	-	10	246,794	10	97,747	12,084
6	0	-	10	266,844	10	144,197	16,413
7	0	-	10	287,307	10	183,976	11,907
8	0	-	10	326,890	10	278,533	18,738

Table 2. Result of optimization of type II function.

	without linkage identification		PICI(LIMS)		LINC-R		
T	#Opt	MNE	#Opt	MNE	#Opt	MNE	STDEV
2	0	-	10	205,808	10	51,417	0
3	0	-	10	252,105	10	51,723	15
4	0	-	10	285,725	10	61,619	11,794

For comparison, the results of GA without linkage identification and PICI are also shown. The results of PICI and the GA without linkage identification are from the work of Tsutsui et al[11]. Standard deviatoins are not shown because they are not reported in the work. Obviously, the proposed method – LINC-R – shows outstanding performance. LINC-R is superior to the conventional methods

because (1) LINC-R has an ability to identify linkage groups precisely on the functions with nonlinearity in a whole domain, and (2) Parallel GA optimizes smaller and easier sub-problems into which the original problem is divided.

4.3 Functions with Nonlinearity in a Part of Search Space

Secondly, we employ the sum of two dimensional trap functions f_D defined by equation (6) which has nonlinearity in a part of its domain.

$$F(\mathbf{x}) = \sum_{i=1}^{n/2} f_D(x_{2i-1}, x_{2i}) \tag{15}$$

This function is to be maximized and has global optimum of $n/2$ at $\mathbf{x} = (0, ..., 0)$. λ is set to 0.8, that means $F(\mathbf{x})$ has a sub-optimum of $0.8n/2$ at $\mathbf{x} = (1, ..., 1)$. Each pair of locus $2i-1$ and locus $2i$ is tightly linked. Thus, F has $n/2$ linkage groups. We set n to 12, 16, and 24. We test LINC-R on the function with various $a(= \pi r^2/4)$ which is the proportion of the nonlinear region on two dimensional f_D. This function gets difficult as a decreases.

For $a = 0.01$, that is the hardest case in this experiment, the population size required for identifying a pair of loci with probability 0.99 is 115 which is obtained by substituting $a = 0.01$ and $Pr = 0.99$ in equation (10). For $n = 24$, the number of function evaluations consumed in the linkage identification stage is 94,964 ($\approx 1.0 \times 10^5$) on that occasion. Thus, in this experiment, linkage identification stage is executed until the number of the function evaluations reached to 1.0×10^5. Then optimization stage is performed until the optimal solution was found or the number of function evaluations reached to 1.0×10^8. 20 runs are performed for each parameter setting.

C_p is set to $10/a$. Since the problem gets difficult as a decreases, more population is required for the problem with smaller a. Thus we set C_p depending on a.

Table 3, 4 and 5 show the number of successful runs which found the optimum (#Opt) and the mean (MNE) and the standard deviation (STDEV) of the number of function evaluations to find the optimum in the runs which found the optimum. The test function F has $n/2$ linkage groups. For LINC-R, the tables also show the average rate of the linkage groups successfully identified (Success rate). For comparison, the result of GA without linkage identification is also shown.

In the case of $n = 12$, LINC-R can obtain optimal solution in all 20 trial runs for every a while so can the GA without linkage identification in a half of trials for $a = 0.02$ and only one trial for $a = 0.01$. In the case of $n = 24$, that is the hardest case, LINC-R can obtain optimum in all trials for a smaller than 0.01 and in 11 trials for $a = 0.01$ while the GA without linkage identification can obtain no optimum even for $a = 0.5$.

LINC-R is more effective in reaching the optimal solutions, however, in the cases of $a = 0.1$ and $a = 0.05$ in $n = 12$ and $n = 16$, it requires higher computational effort than the conventional GA. We thought that is because of an

Table 3. Result of optimization.($n = 12$)

	without linkage identification			LINC-R			
a	#Opt	MNE	STDEV	#Opt	MNE	STDEV	Success rate of linkage identification(%)
0.5	20	255,407	538	20	152,760	5.6	100
0.2	20	291,370	18,777	20	234,831	21,311	100
0.1	20	379,521	24,726	20	513,679	43,283	100
0.05	20	599,305	55,613	20	1,272,980	89,762	100
0.02	10	4,614,750	8,449,032	20	3,679,120	436,108	100
0.01	1	7,769,660	-	20	11,846,500	1,533,870	100

Table 4. Result of optimization.($n = 16$)

	without linkage identification			LINC-R			
a	#Opt	MNE	STDEV	#Opt	MNE	STDEV	Success rate of linkage identification(%)
0.5	20	359,487	11,490	20	187,958	21,375	100
0.2	20	428,539	32,691	20	305,411	6,907	100
0.1	20	617,478	43,013	20	1,208,440	73,266	100
0.05	20	1,095,780	364,831	20	3,441,950	348,240	100
0.02	13	3,737,390	1,210,611	20	5,362,120	440,592	100
0.01	0	-	-	20	18,904,000	2,421,623	100

Table 5. Result of optimization.($n = 24$)

	without linkage identification			LINC-R			
a	#Opt	MNE	STDEV	#Opt	MNE	STDEV	Success rate of linkage identification(%)
0.5	0	-	-	20	208,019	11,559	100
0.2	0	-	-	20	954,755	64,969	100
0.1	0	-	-	20	2,391,770	146,602	100
0.05	0	-	-	20	2,167,270	71,125	100
0.02	0	-	-	20	11,551,660	1,058,316	100
0.01	0	-	-	11	90,454,300	6,519,236	98

inadequate gene exchange schedule. In the optimization stage, optimization is performed in parallel and the population size on island i is set to $|G_i|^2 \times 10/a$ where G_i is the set of loci in linkage group i. Thus, on the problems with smaller a, total population size soars higher. For example, in the case of $n = 12$ and

$a = 0.1$, total population is 2,400. That means a computational effort for 2,400 evaluations is consumed every time genes are exchanged among islands. We believe that LINC-R will be improved more by introducing effective gene exchange schedule.

The average fitness values at the end of trial runs are shown in figure 5. The optimal fitness is 6.0, 8.0, and 12.0 for $n = 12$, $n = 16$, and $n = 24$ respectively. ∘ and △ indicate the average fitness obtained by LINC-R and the GA without linkage identification respectively. As a decreases and the problem gets harder, the performance of the GA without linkage identification inclines while LINC-R hold on to the optimum or near optimum.

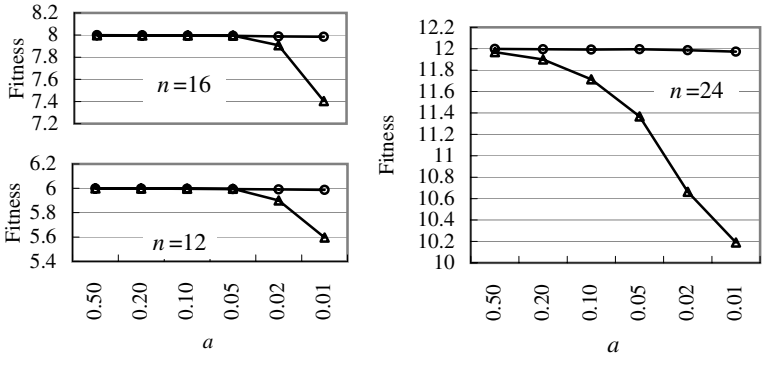

Fig. 5. ∘ shows the average solution obtained by LINC-R and △ shows that by the GA without linkage identification.

5 Conclusion and Discussion

In this paper, we proposed Linkage Identification by Nonlinearity Check for Real-Coded GAs (LINC-R). When an objective function is decomposable to lower order sub-functions, the original problem can be yielded by solving the sub-problems independently and combining them. Linkage identification is a technique to recognize such sub-problems. A decomposed sub-problem is the set of unseparable loci and linear correlated loci are still decomposable. Therefore, the loci in the same linkage group should have nonlinear relation. Testing the equality of two partial differences of an objective function in the domain of the reals, LINC-R detects nonlinearity and reconginzes linkages accurately. The population size required for LINC-R at a certain probability is estimated as a function of the proportion of the nonlinear region in the domain.

It was shown that LINC-R outperformed PICI which identifies linkages according to correlation of loci on Type I and Type II benchmark functions.

Another experiment showed LINC-R with properly sized population had superior performance to the conventional GA without linkage identification on the additively decomposable problems.

If the proposed method is applied to the problems which are not additively decomposable, LINC-R does not detect any linkage groups and optimization is performed by one island. That means the method operates like the conventional GA on such problems except more computational effort is consumed for LINC-R. It is possible that the problems have the other kinds of decomposability such as non-monotonicity and epistasis. Real-world problems have some kinds of decomposablity. For example, a large supply chain can be decomposed to some small supply acitivities. In order to optimize real-world problems, LIMD and LIEM are planned to be introduced in real-valued problems in futur work.

References

1. Harik, G. R. and Goldberg, D. E.: Learning Linkage, In: Belew, R. K. and Vose, M. D. (eds.): Foundations of Genetic Algorithms IV (1996), pp. 247–262.
2. Pelikan, M. Goldberg, D. E., and Cantú-Paz, E.: BOA: The Bayesian Optimization Algorithm, Proc. the 1999 Genetic and Evolutionary Computation Conference (1999), 525-532.
3. Munetomo, M. and Goldberg, D. E.: Designing a genetic algorithm using the linkage identification by nonlinearity check, IlliGAL Technical Report (1998).
4. Munetomo, M. and Goldberg D. E.: Linkage Identification by Non-monotonicity Detection for Overlapping functions, Evolutionary Computation, vol. 7, no. 4 (1999), pp. 377–398.
5. Munetomo, M.: Linkage identification based on epistasis measures to realize efficient genetic algorithms, Proc. the 2002 Congress on Evolutionary Computation (2002).
6. Fogel, D. B.: Real-valued vectors, In: Bäck, T., Fogel, D. B. and Michalewicz, Z. (eds.): Handbook of Evolutionary Computation (1997), C1.3:1–2.
7. Wright, A. H.: Genetic Algorithms for Real Parameter Optimization, In: Rawlins, G. J. E. (ed.): Foundations of Genetic Algorithms (1991).
8. Michalewicz, Z. and Janikow, C. Z.: Handling Constraints in Genetic Algorithms, Proc. 4th Intl. Conf. on Genetic Algorithms (1991), pp. 151–157.
9. Eshelman, L. J. and Schaffer, J. D.: Real-Coded Genetic Algorithms and Interval-Schemata, In: Whitley, L. D. (ed.): Foundations of Genetic Algorithms 2 (1993).
10. Tsutsui, S., Goldberg, D. E., and Sastry, K.: Linkage Learning in Real-Coded GAs with Simplex Crossover, Proc. 5th Intl. Conf. Artificial Evolution (2001), pp. 73–84.
11. Tsutsui, S. and Goldberg, D. E.: Simplex Crossover and Linkage Identification: Single-Stage Evolution VS. Multi-Stage Evolution, Proc. the 2002 Congress on Evolutionary Computation (2002), pp. 974–979.
12. Higuchi, T., Tsutsui, S., and Yamamura, M.: Theoretical analysis of simplex crossover for real-coded Genetic Algorithms, Proc. PPSN VI (2000), pp. 365–374.
13. Sato, H., Yamamura, M., and Kobayashi, S.: Minimal generation gap model for GAs considering both exploration and exploitation, Proc. the IIZUKA'96 (1997), pp. 494–497.