Progress Toward Linkage Identification in Real-Coded GAs with Simplex Crossover

Shigeyoshi Tsutsui, David E. Goldberg, and Kumara Sastry

IlliGAL Report No. 2000033
October 2000

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue
Urbana, Illinois 61801 USA
Phone: 217-333-0897
Fax: 217-244-5705
Web: http://www-illigal.ge.uiuc.edu/
Progress Toward Linkage Learning in Real-Coded GAs with Simplex Crossover

Shigeyoshi Tsutsui, David E. Goldberg, and Kumara Sastry

Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
117 Transportation Building,
{shige, deg, kumara}@illigal.ge.uiuc.edu

Abstract
In recent years, many researchers have concentrated on using real-valued genes in genetic and evolutionary algorithms (GEAs). Previous studies have proposed simplex crossover (SPX) for real-coded GAs. SPX has several good characteristics and works well on various test functions. However, SPX fails on functions that consist of tightly linked sub-functions. On those functions, SPX should be applied on each tightly linked group of parameters. In this paper, we propose a method of linkage identification in real-coded GAs with SPX and evaluate it using several test functions. The mechanism works well on many of the test functions used. We also discuss difficulties with the proposed method on more complex test functions and show possible solutions to the problems.

1. Introduction
In genetic and evolutionary algorithms (GEAs), evolution strategies (ESs) and evolutionary programming (EP) traditionally use real value vectors to represent a problem's structure. In contrast to this, genetic algorithms (GAs) traditionally use bit string representation. However, in recent years many researchers have concentrated on using real-valued genes in GAs [Davis, 91], [Wright 91], [Janikow 91], [Surry 96], [Ono 97, 99]. Theoretical studies of real-coded GAs have also been performed [Goldberg 91a], [Crossman 92], [Eshelman 93], [Qi 94a, b], [Kita, 99] [Higuch, 00].

Previous studies [Tsutsui 99], [Higuchi 00] have proposed simplex crossover (SPX) for real-coded GAs. Let \( n \) be the number of parameters. Then, \( n+1 \) parental vectors form a simplex in \( \mathbb{R}^n \). SPX generates offspring by uniformly sampling points inside the simplex formed by \( n+1 \) parental vectors. SPX has various good characteristics, e.g., it does not depend on a coordinate system, the mean vector of parents and offspring generated with SPX are the same, and SPX can preserve a covariance matrix of the population with an appropriate parameter setting. As a result, SPX works well on various test functions [Higuchi 00]. However, SPX fails on functions that consist of multiple tightly linked sub-functions. On those functions, SPX should be applied on each tightly linked parameters group, i.e., each simplex should be formed in each subspace in which parameters are tightly linked. Thus, we need a method of identifying those tightly linked parameter groups.

There have been many studies of linkage identification in binary-coded GAs to find useful building blocks.
(BBs) [Goldberg 89], [Goldberg 93], [Lobo 98], [Harik 99], [Perikan 99], [Munetomo 99]. However, there exist few
studies on linkage identification in real-coded GAs. In this paper, we propose a method of linkage identification
for real-coded GAs with SPX and evaluate it using several test functions. The results shows that it works fairly
well on the test functions used. We also discuss some of the problems involved with the proposed method on
more complicated test functions and propose possible solutions to the problems.

In the remainder of this paper we briefly review SPX; in Section 3, we propose a linkage identification
method for real-coded GAs with SPX; in Section 4, empirical results are given; problems with the proposed
method and future work are discussed in Section 5. Finally, concluding remarks are made in Section 6.

2. A Brief Review of SPX

This section gives a brief review of SPX [Tsutsui 99], [Higuchi 00].

2.1 Definition of SPX

The SPX operator uses \(n+1\) parental vectors \(X_i, i = 0, 1, ..., n\) for recombination. These \((n+1)\) vectors form a
simplex in \(R^n\). Then this simplex is expanded in each direction \((X_i - O)\) to some extent, where \(O\) is the center of
mass of \((n+1)\) parental vectors. Offspring are then generated by uniformly picking vector values from this
expanded simplex.

The formal procedure of SPX for generating an offspring from \((n+1)\) parents can be written as follows:

**Step 1:** Choose \((n+1)\) parental vectors \(X_k (k=0, 1, ..., n)\) according to the generational model used and calculate
their center of mass \(O\) as

\[
O = \frac{1}{n+1} \sum_{k=0}^{n} X_k
\]

**Step 2:** Generate random numbers \(r_i\) as

\[
r_k = u \frac{1}{k+1}, \quad k = 0, 1, ..., n
\]

where \(u\) is uniform random number \(\in [0.0, 1.0]\).

**Step 3:** Calculate \(Y_i\) and \(C_i\), respectively, as

\[
Y_k = O + \varepsilon (X_k - O), \quad k = 0, 1, ..., n
\]

\[
C_k = \begin{cases} 0 & : k = 0 \\ r^{(k-1)} (Y_{k-1} - Y_k + C_{k-1}) & : k = 1, 2, ..., n \end{cases}
\]

where \(\varepsilon\) is the expansion rate, a control parameter of SPX.

**Step 4:** Generate an offspring \(C\) as

\[
C = Y_n + C_n
\]

Figure 1 shows SPX in \(R^2\). In \(R^3\), three parental vectors, \(X_0, X_1, \) and \(X_2\) are used.

2.2 Statistical properties of SPX

The SPX has the following statistical properties [Higuchi 00].

(1) SPX does not depend on a coordinate system, i.e., if other GA operators such as a selection operator do not
depend on the coordinate system, the performance of GAs with SPX is independent of linear coordinate transformation.

(2) The mean vector of parents \(<P>\) and the mean vector of offspring generated with SPX \(<C>\) are the same, i.e., \(<P> = <C>\).

(3) Let \((m + 1)\) be a number of parents. A covariance matrix of distribution of offspring generated with SPX are given by

\[
\{\psi_{ij}^C\} = \frac{1}{m+1} \left[ 1 + \varepsilon^2 \frac{m}{m+2} \right] \{\psi_{ij}^P\}
\]  

(6)

where, \{\psi_{ij}^P\} is a covariance matrix of distribution of parents.

In Eq. (6), if we take the expansion rate as

\[
\varepsilon = \sqrt{m + 2}
\]  

(7)

then,

\[
\{\psi_{ij}^C\} = \left( \frac{1}{\sqrt{m+2}} \right)^2 \{\psi_{ij}^P\}
\]  

(8)

In other word, SPX preserves the covariance matrix of the population. The value of expansion rate given by Eq. (7) is a theoretical guideline in using SPX and it has been shown that the performance of SPX with this value works well on various test functions [Higuchi 00].

Recently, a number of evolutionary algorithms that guide the exploration of search space by building probabilistic models of promising solutions found so far (PMBGAs; the probabilistic model-building GAs) have been proposed, and these algorithms have been shown to perform very well on a wide variety of problems [Pelikan 00]. In PMBGAs, a model (an estimate of true distribution) of promising solutions must be constructed and new solutions are then generated according to the constructed model. In GAs with SPX, we do not need to construct any models since SPX can preserve a covariance matrix in generating offspring if we use the expansion rate given by Eq. (7). As a result SPX implicitly constructs a model of a current population in generating offspring.
3. Linkage Identification with SPX

In this section, we consider functions which have nonlinear subsets of variables and seek tight linkage of variables in the nonlinear subfunction.

### 3.1 Division of SPX into sub-spaces

In this study we consider evaluation functions that can be written as

\[ F(X) = F \text{tight}_1(X \text{tight}_1) + \cdots + F \text{tight}_s(X \text{tight}_s) + F \text{loose}(X \text{loose}) \]  

(9)

where, \( X_{\text{tight}, 1} \cap \cdots \cap X_{\text{tight}, s} \cap X_{\text{loose}} = \emptyset \).

Here, each sub-function \( F_{\text{tight}_s}(X_{\text{tight}_s}) \) \((s=1, \ldots, S)\) has tight linkage among parameters and \( F_{\text{loose}}(X_{\text{loose}}) \) has weak or no linkage among parameters (of course we may assume there is no sub-function \( F_{\text{loose}}(X_{\text{loose}}) \)). We denote the number of parameters in \( X_s \) as \( |X_s| \) and call it the order of \( X_s \). On these functions, SPX in \( \mathbb{R}^n \) does not work well as shown in Section 4. For these kinds of functions, we intuitively notice that it is better to apply SPX in each subspace \( X_{\text{tight}, s} \) \((s=1, \ldots, S)\) and \( X_{\text{loose}} \) separately (Figure 2). To assume that a given function is separable into several sub-functions may seem like a special case in general optimization tasks. But this form should work for quasi-separable functions, i.e., where there is some crosstalk but crosstalk is weak. Here we note that each parameter set \( X_j \) corresponds to a building block (BB) in binary string GAs.

![Figure 2: Division of SPX](image)

### 3.2 Linkage identification methods

For a given optimization problem, we usually do not know the linkage sets \( X_j \) in \( X \subset \mathbb{R}^n \). Thus, we need to consider methods of linkage identification.

Linkage identification methods in binary string (or finite alphabet) GAs have been studied intensively. In binary GAs, preserving useful building blocks (BBs) -- essential sub-components of solutions -- and mixing them through crossover and selection operations are essential [Goldberg 89]. For effective mixing, a set of loci
that belongs to a BB needs to be tightly linked in crossover to avoid distribution. Here, the tightness of a loci
is referred to as linkage and a set of loci tightly linked is called a linkage set or linkage group [Munetomo 99].
Thus, the identification of linkage groups becomes very important for developing effective GAs.

In real-coded GAs, the concept of linkage is basically identical with linkage in binary-coded GAs, in the
sense that each linkage group in real parameters is essentially a sub-component of a solution and parameters
in the same linkage group are mutually correlated on the fitness landscape.

Previous studies on linkage group identification in binary GAs can be classified into the following three
categories [Munetomo 99]:
1. Direct detection of bias in probability distribution
2. Direct detection of fitness changes by perturbations
3. Indirect detection along genetic search of BBs

In this study, we propose a method that falls under the first category of linkage identification. This is
described in the next subsection in detail.

### 3.3 Linkage identification with piecewise interval correlation by iteration

Consider the case where the evaluation function $F(X)$ is non-negative. Then the probability density function
(p.d.f.) $p(X,t)$ of individuals in the population under proportional selection at generation $t$ can be written as

$$p(X,t) = \frac{F^t(X) p_0(X)}{\int F^t(X) p_0(X) dX}$$

(10) [Goldberg 91b], where $p_0(X)$ is an initial p.d.f. of individuals and no special bias is assumed to be added by
crossover and mutation operators. Normally $p_0(X)$ is the uniform distribution. Then, $p(X,t)$ can be written as

$$p(X,t) = \frac{F^t(X)}{\int F^t(X) dX}$$

(11)

From Eq. (11), we can see that the distribution of individuals in a population reflects the fitness landscape
(evaluation function) $F(X)$ and this reflection is amplified as generation $t$ increases by the term $F^t(X)$. Thus as
generation proceeds, more individuals are concentrated in the portions where there exist individuals with
higher evaluation functional values. This discussion holds true for other selection operations such as ranking
selection or tournament selection, where individuals which have higher functional values have a higher chance
to be selected for next generations.

We notice from the above discussion that the linkage information among parameters can be obtained by
observing the distribution of individuals in a population. If $F(X)$ has a linkage among parameters on loci $x_i$, $x_j$, and $x_k$, then there should be some degree of correlation between $x_i-x_j$, $x_j-x_k$, and $x_k-x_i$. Thus if we examine the
correlation coefficient matrix $R = [\rho_{ij}]$ of parameter values of individuals in a population, linkage among
parameters might be detected.

Let $x_i = \{x_{i1}, x_{i2}, ..., x_{in}\}$ and $x_j = \{x_{j1}, x_{j2}, ..., x_{jn}\}$ be parameter values of individuals in the population on loci $x_i$ and $x_j$, respectively. Then the correlation coefficient $\rho_{ij}$ can be calculated as
\[ \rho_{ij} = \frac{1}{N} \sum_{p=1}^{N} (x_{i,p} - \bar{x}_i)(x_{j,p} - \bar{x}_j) \]

where \( \bar{x}_i \) and \( \bar{x}_j \) are mean values of \( x_i \) and \( x_j \) respectively, and \( V(x_i) \) and \( V(x_j) \) are variances of \( x_i \) and \( x_j \) respectively. Since this normal correlation coefficient can examine only linear correlations among parameters, we must use some non-linear estimation technique such as non-linear regression or higher moment methods.

In this paper, we propose a piecewise interval correlation by iteration (PICI) algorithm, a more simple and straightforward extension of Eq. (12). The principle of PICI algorithm is simple. It calculates correlation coefficients of piecewise intervals. The PICI algorithm is shown schematically in Figure 3. In PICI, to obtain \( \rho_{ij} \) we first divide the \( x_i, x_j \) space into four equal squares (quadrants). This division is recursively repeated until we get small squares such that if we divide those small squares into quadrants, then each of the squares in the quadrants has fewer individuals than some specified minimum \( N_{\text{min}} \). Interval correlation coefficients are calculated for all and only quadrants in which each of the squares within the quadrant(s) has fewer individuals than the minimum \( N_{\text{min}} \). We then take the weighted average of these four correlation coefficient values. Figure 4 shows a simple example of PICI algorithm where a value of \( N_{\text{min}} \) is assumed. The piecewise interval correlation coefficients are calculated inside of each square surrounded with thick segments and the weighted average of each absolute value is taken as the correlation coefficient between \( x_i \) and \( x_j \).

### 3.4 SPX with linkage identification

Let \( R(t) = [\rho_{ij}(t) ] \) be the correlation coefficient matrix obtained by PICI algorithm at generation \( t \). Then a linkage group \( X_{\text{tight},s} \) in Eq. (9) at generation \( t \) can be obtained by using the following properties: if \( x_i \) is a member of a linkage group \( X_{\text{tight},s} \) then \( x_i \) must be satisfied

\[ x_i \in X_{\text{tight},s} | s \in \{1, ..., S\} \text{ iff } \{ \rho_{ij}(t) \geq K_{\rho} \ | \forall x_j \in X_s \} \land \{ \rho_{ij}(t) < K_{\rho} \ | \forall x_j \not\in X_s \}, \]

where \( K_{\rho} \) is the threshold value to determine the existence of linkage. \( X_{\text{loose}} \) is obtained as

\[ X_{\text{loose}} = X - \bigcup_{s=1}^{S} X_{\text{tight},s} \]

Figure 5 shows an example of linkage groups developed from a correlation coefficient matrix. Each node represents a parameter (locus) and each arc represents a correlation value. In general, estimated correlation coefficient \( R(t) = [r_{ij}(t)] \) at generation \( t \) includes statistical errors and the value may vary as generation proceeds. To reduce this variation, we use the following weighted estimation:

\[ \rho_{ij}(t) = (1-\lambda)\rho_{ij}(t-1) + \lambda \rho'_{ij}(t) \]

where \( \rho'_{ij}(t) \) is the observed correlation coefficient by PICI at generation \( t \) and \( \rho_{ij}(t) \) is the estimated correlation coefficient. \( \lambda \) (0 < \( \lambda \) ≤ 1) is a constant parameter for estimation.

SPX is applied to obtained linkage groups observed at generation \( t \) in each GA run. Figure 6 shows how to apply linkage identification information \( R(t) = [\rho_{ij}(t)] \) to the SPX operator. First we apply SPX on \( X \). When
Figure 3: Piecewise interval correlation by iteration (PICI) algorithm

Figure 4: A simple example of PICI in $x_i, x_j$ space
\[ X = \{x_1, x_2, x_3, x_4, x_5, x_6\} \]
\[ X_{\text{tight,1}} = \{x_1, x_2\}, \quad X_{\text{tight,2}} = \{x_3, x_4\}, \quad X_{\text{loose}} = \{x_5, x_6\} \]

Figure 5: Linkage identification from \( R(t) \). \( K_p \) value of 0.2 is assumed

Figure 6: SPX with linkage identification
linkage groups by Eq. 13 are observed, we remove those groups from $X$ and apply SPX separately on those groups. Thus linkage groups are identified as evolution proceeds. Of course we allow for the collapse of detected linkage groups.

4. Empirical Study

To evaluate the linkage identification method proposed in Section 3, we run a real-coded GA. The experimental methodology, test functions used, and experimental analysis are described in this section.

4.1 Experimental methodology

4.1.1 Evolutionary model

The basic evolutionary model we used in these experiments are based on minimal generation gap (MGG) proposed in [Satoh 97]. The MGG model has the desirable convergence property of maintaining the diversity of the population, and performs better than the other conventional models in a wide range of applications. We extended MGG to SPX as follows (Figure 7):

1. Set generation counter $t = 0$. Generate $N$ individuals randomly as an initial population $P(t)$.
2. Apply piecewise interval correlation by iteration (PICI) algorithm to population $P(t)$ and get the correlation coefficient matrix $R(t)$ and detect linkage groups based on Eq. 13.
3. Select a set of $m$ (= $n+1$; $n$ is the number of parameters) parents $M$ by random sampling from the population $P(t)$.
4. Generate a set of $m \times F_{MGG}$ offspring $C$ by applying the SPX to $M$ based on linkage identification. We used $F_{MGG}$ value of 10.
5. Select a set of $m$ individuals $NEW$ from $M+C$ using tournament selection and replace $M$ with $NEW$ in population $P(t)$ and get population $P(t+1)$.
6. Stop if specified conditions are satisfied, otherwise set $t = t+1$ and go to step 2.

Figure 7: Evolutionary model
4.1.2 Performance measure

We evaluated the algorithms by measuring their #OPT or number of runs in which the algorithm succeeded in finding the global optimum, and MNE or mean number of function evaluations to find the global optimum in those runs where it did find the optimum. We used the value $\Delta x_j$ as a resolution to determine whether the optimal solution was found. We defined the successful detection of the solution as being within $\Delta x_j$ range of the actual optimum point. Let $(o_1, ..., o_n)$ be the optimal solution of a function. If all parameters $(x_1, ..., x_n)$ of the best individual are within the range $[(o_j - \Delta x_j/2), (o_j + \Delta x_j/2)]$ for all $j$, we assumed the real-coded GA to have found the optimal solution. Twenty (20) runs were performed in each experiment. In each run, the initial population $P(0)$ was randomly initialized in the search space. Each run continued until the global optimum was found or a maximum of 2,000,000 function evaluations was reached. A population size of 500 and $K_\rho$ value of 0.25 for linkage identification were used for all experiments.

4.2 Test functions and results

In this paper, we used two types of test functions. One type (Type I) is a function which consists of a tight linked sub-function $F_{\text{tight}}(X_{\text{tight}})$ and a sub-function $F_{\text{loose}}(X_{\text{loose}})$ with weak or no linkage among parameters. The other type of function (Type II) consists of multiple sub-functions with tight linkage among parameters of order 2.

4.2.1 Type I functions

Here we consider the following function:

$$F_{\text{TL}}(X, Y) = F_{\text{tight}}(X_{\text{tight}}) + F_{\text{loose}}(Y_{\text{loose}}) = F_{\text{RT}}(x_1, ..., x_T) + F_{\text{SL}}(y_1, ..., y_L) - 2.048 \leq x_i, y_j < 2.047$$

(16)

where, $F_{\text{RT}}(x_1, ..., x_T)$ is a $T$-dimensional Rosenbrock function defined as

$$F_{\text{RT}}(x_1, ..., x_T) = \sum_{i=2}^{T} [100(x_i - x_{i-1}^2)^2 + (x_i - 1)^2]$$

(17)

and has tight linkage among parameters $x_1, ..., x_T$. $F_{\text{SL}}(y_1, ..., y_L)$ is an $L$-dimensional Sphere function defined as

$$F_{\text{SL}}(y_1, ..., y_L) = \sum_{j=1}^{L} (y_j - 1)^2$$

(18)

and has no linkage among parameters $y_1, ..., y_L$. The total dimension of this function $n$ is $(T + L)$. The global solution is at $(1, ..., 1)$. The precision $\Delta x_j, \Delta y_j$ of solution of this function is defined as 0.001.

To see how the linkage groups are identified during evolution, let us observe a change of correlation coefficient $R(t)$ on function $F_{\text{TL}}(X, Y) = F_{\text{RT}}(x_1, x_2) + F_{\text{SL}}(y_1, y_2)$. On this function, there should be a tight linkage between $x_1$ and $x_2$ and there should be no linkage among other parameters. Figure 8 shows the change of $R(t)$ on $F_{\text{TL}}(X)$ until an optimal solution was found with SPX on $(X, Y)$ in a typical single run. No linkage identification is applied to SPX. Figure 9 shows the distribution of individuals at 20,000 function evaluations in: (a) $x_1, x_2$ space; (b) $x_1, y_1$ space; and (c) $y_1, y_2$ space, respectively. We can see that $R(t)$ reflects linkage among parameters correctly.
Figure 8: Correlation coefficient on function $F_{n}(X)$

Figure 9: Distribution of individuals at 20,000 function evaluations on function $F_{n}(X)$ in: (a) $x_1, x_2$ space; (b) $x_1, y_1$ space; and (c) $y_1, y_2$ space

Table 1 Results on function $F_{TL}(X)$

<table>
<thead>
<tr>
<th>No of parameters</th>
<th>without linkage learning</th>
<th>with linkage learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>#OPT</td>
<td>MNE</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>207,479.1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>590,153.4</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>441,632.0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1,760,400.0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>--</td>
</tr>
</tbody>
</table>
Now, let us fix \( L \), dimension of function \( F_{3}(y_{1},...y_{r}) \), be 20 and run for \( T = 2, ..., 8 \) (dimension of function \( F_{RF}(x_{1}, ..., x_{p}) \) with and without linkage identification. Table 1 shows the result of this experiment. Without linkage identification, \( ^{\#OPT} = 20 \) only for \( T = 2 \). For \( T = 3, 4, \) and \( 5, \) \( ^{\#OPT} = 8, 1, \) and \( 1 \), respectively. For \( T > 6, \) \( ^{\#OPT} = 0 \). On the other hand, with linkage identification, \( ^{\#OPT} = 20 \) for \( T = 2 \) to 6. For \( T = 7 \) and 8, \( ^{\#OPT} = 16 \) and 15, respectively. There were far fewer MNTs with linkage identification than without linkage identification.

Figure 10 shows changes of functional values of best individuals in populations in 20 runs for \( T = 2, 4, 6, \) and 8. In each run, once linkage of \( F_{RF} \) was identified correctly, GA converged to the optimum rapidly. Although as \( T \) increased, MNE also became larger and the algorithm failed to identify linkage correctly 4 and 5 times in 2,000,000 function evaluations for \( T = 7 \) and 8 respectively, but we can see that the linkage identification scheme is working fairly well on this function.

![diagram](image)

(a) \( T = 2 \)  
(b) \( T = 4 \)  
(c) \( T = 6 \)  
(d) \( T = 8 \)

**Figure 10**: Convergence processes on function \( F_{TL}(X) \) for \( T = 2, 4, 6, \) and 8

### 4.2.2 Type II functions with order 2

In this subsection, we use the following two test functions which have multiple tight linkage groups with order 2 as follows:

\[
F_{R2-n}(X) = F_{R2}(x_{1},x_{2}) + F_{R2}(x_{3},x_{4}) + ... + F_{R2}(x_{n-1},x_{n})
\]

(19)

\[
F_{D2-n}(X) = F_{D2}(x_{1},x_{2}) + F_{D2}(x_{3},x_{4}) + ... + F_{D2}(x_{n-1},x_{n})
\]

(20)

In these functions, we do not consider \( F_{loose}(X_{loose}) \) of Eq. (9). Function \( F_{R2-n}(X) \) is an \( n \)-parameter function and
is the sum of \( n/2 \) Rosenbrock sub-function \( F_{R2} \) (well known \( F2 \) of De Jong’s test suite). Sub-function \( F_{R2} (x_1, x_2) \) is defined as follows:

\[
F_{R2} (x_1, x_2) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2, \quad -2.048 \leq x_1, x_2 < 2.047.
\]

This sub-function has a tight linkage between parameters \( x_1 \) and \( x_2 \). The resolution \( \Delta x_i \) of this function is defined as 0.001.

Function \( F_{D2}(X) \) is also an \( n \)-parameter function and is the sum of \( n/2 \) sub-function \( F_{R2} \). Sub-function \( F_{D2} (x_1, x_2) \) is defined as follows:

\[
F_{D2} (x_1, x_2) = \begin{cases} 
1 - r / k & : \text{if } 0 \leq r < k \\
- h k / (10 - k) + h r / (10 - k) & : \text{if } k \leq r < 10 \\
h (20 - k) / (10 - k) - h r / (10 - k) & : \text{if } 10 \leq r < 20 - k \\
0 & : \text{otherwise.}
\end{cases}
\]

where, \( r = \sqrt{x_1^2 + x_2^2} \). This sub-function has a strong deceptive attractor around \( r = 10 \) as shown in Figure 11 and thus has a tight linkage between parameters \( x_1 \) and \( x_2 \). We use \( k \) value of 1 and \( h \) value of 0.8. The resolution \( \Delta x_i \) of this function is defined as 0.01.

![Figure 11: Function \( F_{D2}(x_1, x_2) \)](image)

(1) Results on \( F_{R2_{-n}}(X) \)

To see how linkage groups are identified during the evolution, let us observe the change of correlation coefficient \( R(t) \) on function \( F_{R2_{-n}}(X) = F_{R2_{-n}}(x_1, x_2) + F_{R2_{-n}}(x_3, x_4) \). On this function, there should be tight linkages between parameters \( x_1 \) and \( x_2 \), and between \( x_3 \) and \( x_4 \), respectively. There should be no linkage among other parameters. Figure 12 shows the change of \( R(t) \) on \( F_{R2_{-n}}(X) \) with SPX on \( X \) in a typical single run. No linkage identification is applied to SPX. Figure 13 shows the distribution of individuals at 20,000 function evaluations: (a) on \( x_1, x_2 \) space; and (b) on \( x_1, x_3 \) space, respectively. Thus we can confirm the \( R(t) \) reflects linkage among parameters correctly.

Table 2 shows the performance for \( n = 4, 8, ..., 40 \) both with and without linkage identification. With linkage identification, SPX worked very well up to \( n = 40 \). Without linkage identification SPX failed to find the optimal solution for \( n \geq 16 \), and MNEs without linkage identification were much larger than MNE with linkage identification for \( n = 4, 8, \) and 12. Fig 14 shows processes of linkage identification in typical single runs until the optimum solution was found for \( n = 4, ..., 40 \). We can observe that linkage groups are identified correctly as evolution proceeds.
Figure 12: Correlation coefficient on function $F_{R2-4}(X)$

Figure 13: Distribution of individuals at 20,000 function evaluations on $F_{R2-4}(X)$ in:
(a) $x_1, x_2$; and (b) $x_1, x_3$

Table 2 Results on function $F_{2-n}(X)$

<table>
<thead>
<tr>
<th>No of parameters</th>
<th>without linkage learning</th>
<th>with linkage learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>#OPT</td>
<td>MNE</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>67,721.8</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>191,046.8</td>
</tr>
<tr>
<td>12</td>
<td>18</td>
<td>697,515.4</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>--</td>
</tr>
</tbody>
</table>

15
(2) Results on $F_{D2-n}(X)$

To see how the linkage groups are identified during the evolution, let us observe the change of correlation coefficient $R(t)$ on function $F_{D2-4}(X)=F_{D2}(x_1,x_2)+F_{D2}(x_3,x_4)$ again. On this function there should be tight linkage between parameters $x_1$ and $x_2$, and between $x_3$ and $x_4$, respectively, at deceptive attractor around $r = 10$. There should be no linkage among other parameters. Figure 15 shows the change of $R(t)$ on $F_{D2-4}(X)$ until 10,000 function evaluations. No linkage identification is applied to SPX. Figure 16 shows the distribution of individuals at 20,000 function evaluations in: (a) $x_1$-$x_2$ space; and (b) $x_3$-$x_4$ space, respectively. As on function $F_{2-n}(X)$, we can confirm the $R(t)$ reflect linkage among parameters correctly.

Table 3 shows the performance for $n = 4, 6, 8, 10$ both with and without linkage identification. Function $F_{D2-n}(X)$ is much more difficult to solve compared with function $F_{2-n}(X)$ since this function has a strong attractor. With linkage identification, the algorithm found the optimum solution for $n = 4$ (#OPT=20, MNE = 143,054.0), 6 (#OPT=11, MNE = 233,660.0), and 8 (#OPT=2, MNE = 283,995.0). No optimal solution was found for $n > 8$. However, the performance with linkage identification is better than that without linkage identification.

5. Discussion and Future Work

In this section, we review the results in Section 4 and discuss problems involved with the proposed method and possible solutions.

5.1 Problems of proposed method

5.1.1 Why did linkage identification fail for larger $T$ on function $F_{TL}(X)$?

For $T = 8$ on function $F_{TL}(X)$, #OPT = 15 and five (5) runs failed to reach the optimum solution in 2,000,000
Figure 15: Linkage identification from correlation coefficient

Figure 16: Distribution of individuals at 20,000 function evaluations on $F_{D2-n}(X)$ in:
(a) $x_1, x_2$; and (b) $x_1, x_3$

Table 3 Results on function $F_{D2-n}(X)$

<table>
<thead>
<tr>
<th>No of parameters</th>
<th>Without linkage learning</th>
<th>With linkage learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>#OPT</td>
<td>MNE</td>
</tr>
<tr>
<td>4</td>
<td>20</td>
<td>300,823.0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>--</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>--</td>
</tr>
</tbody>
</table>
function evaluations. Figure 17 (a) shows the change of correlation coefficient of the 4th run that found the optimum solution at 287,319 function evaluations. To make the figure simple, the correlation coefficient is shown only between $x_i - x_j$ ($i = 2, ..., 8$) and $x_i - y_j$ ($j = 1, ..., 20$). But we can see that the linkage identification was performed well in this run.

Figure 17 (b) shows the change of correlation coefficient of the 11th run that did not find the optimum solution in 2,000,000 function evaluations. Again, the correlation coefficient is shown only between $x_i - x_j$ ($i = 2, ..., 8$) and $x_i - y_j$ ($j = 1, ..., 20$). We can observe that the correct linkage identification was not performed with this run. Figure 18 shows the change of correlation coefficient in the same pairs of loci, with the same initial distribution of individuals, and without linkage identification, i.e., SPX is applied on $(X, Y)$. In this figure, we can observe several **spurious correlation coefficient** values that exceeded the threshold value $K_\rho (0.25, \text{see 4.1.2})$. These could have resulted in forming incorrect linkage groups, which would have then been applied to SPX. This may explain why incorrect correlations shown in Figure 17 (b) occurred, and why the optimum solution was not found in the 11th run.

5.1.2 What are the results for functions which have multiple tight linkage groups with order 3?

Results on function $F_{R2-n}(X)$ with linkage identification in Section 4 were good and we found the linkage identification was performed fairly well. What are the results for functions which have multiple tight linkage groups with order 3? Table 4 shows results on function $F_{R3-n}(X)$, which consists of 3-dimensional Rosenbrock sub-functions. With this function, the performance was much worse.

Figure 19 (a) shows the change of correlation coefficient of the first run on function $F_{R3-15}(X)$ which could not find the optimum solution in 2,000,000 function evaluations. The correlation coefficient is shown only between $x_i$ and $x_j$ ($i = 2, ..., 15$) in this figure, but we can observe that the correct linkage identification was not performed with this run. Figure 19 (b) shows the change of correlation coefficient between the same pairs of loci, with the same initial distribution of individuals, and without linkage identification, i.e., SPX is applied on $X$. In this figure, we can observe several big **spurious correlation coefficient** values at the early stage of the evolution. These would result in forming incorrect linkage groups in this function, too.

5.2 Possible solutions

Discussion in Subsection 5.1 suggests several possible solutions to cope with the problems of the current method. The main problem with the current method is that the linkage identification is performed with high sensitivity to values of the correlation coefficient matrix $R(t)$ at each generation. Here we may propose two possible ways of reducing this hypersensitivity: (1) **linkage identification with multi-stage evolution (LIME)**; and (2) **linkage identification with offline evolution (LIOF)**, to reduce this over sensitivity.

(1) Linkage identification with multi-stage evolution (LIME)

In the linkage identification with multi-stage evolution (LIME) approach, the evolution process is divided into multiple stages. Figure 20 shows a typical case where the evolution process is divided into three stages -- the initial, learning, and searching stages. In the initial stage, we evolve without applying any reflection of the identification information to SPX. In this stage, the population begins to form a linkage structure. In the learning stage, we observe the correlation coefficient $R(t)$ by PICI algorithm and accumulate the values. In this stage we do not reflect any linkage information to SPX, i.e., as in initial stage, SPX is applied to one group $X$. At
the end of this stage, we analyze the accumulated \( R(t) \) and identify linkage groups. This information on linkage groups is passed to the succeeding searching stage. In the searching stage, SPX is applied to each identified linkage group in the previous stage and searching is performed.

(2) **Linkage identification with offline evolution (LIOF)**

In the linkage identification with offline evolution (LIOF) approach, we first run a linkage identification GA (LIGA). In LIGA, we only observe and accumulate the correlation coefficient \( R(t) \) values during its run without applying any linkage identification. After the end of this GA, we analyze the set of accumulated \( R(t) \) data and identify linkage groups. Then, the searching task GA (STGA) follows. It uses linkage groups information identified based on LIGA (see Figure 21).
Table 4 Results on function $F_{R3-n}(X)$

<table>
<thead>
<tr>
<th>No of parameters</th>
<th>without linkage learning</th>
<th>with linkage learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>#OPT</td>
<td>MNE</td>
</tr>
<tr>
<td>6</td>
<td>20</td>
<td>126,791.3</td>
</tr>
<tr>
<td>9</td>
<td>20</td>
<td>250,922.5</td>
</tr>
<tr>
<td>12</td>
<td>12</td>
<td>707,995.1</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>--</td>
</tr>
</tbody>
</table>

Figure 19: Correlation coefficient for $n = 15$ in $x_i, x_i$ ($i = 2, ..., 15$) on function $F_{R3-n}(X)$: (a) with linkage identification; and (b) without linkage identification

Figure 20: Linkage identification with multi-stage evolution (LIME)

Figure 21: Linkage identification with multi-stage evolution (LIMS)
5.3 Future work

Following the conclusion of the main results in Section 4, we tried a pilot study of LIME on function $F_{TL}(X,Y)$ with $T = 8$. Results in Figure 18 shows $R(t)$ becomes stable after 100,000 function evaluations. So, we used $t_{\text{begin}}$ value of 100,000 and $t_{\text{end}}$ value of 120,000 in the function evaluation counter. To identify linkage groups in this pilot study, we simply took the average value of $R(t)$ observed in the learning stage. Results are promising as shown in Figure 22. #OPT was 20, and MNE was 317,453.7. We also tried this approach on function $F_{R312}(X)$ with promising result (#OPT = 20, MNE = 397,518.3).

Although we used a fixed-size population for all test problems in this study, population sizing should correlate strongly with problem complexity [Harik 96]. This must also be studied.

More work is needed but we are hopeful that LIME and/or LIOF will be reliable over all the test cases herein.

6. Conclusions

In binary string GAs, linkage identification methods have been intensively studied. Preserving useful building blocks (BBs) and mixing them through crossover and selection operations are essential. For effective mixing, a set of loci that belongs to a BB needs to be tightly linked in crossover to avoid distribution. Thus, the identification of linkage groups becomes very important for developing effective GAs. In real-coded GAs, the concept of linkage is basically identical with linkage in binary-coded GAs, in the sense that each linkage group in real parameters is essentially a sub-component of a solution, and parameters in the same linkage group are mutually correlated on the fitness landscape.

In this paper, we have proposed a method for linkage identification in real-coded GAs with simplex crossover (SPX). In GAs with appropriate selection operators, the distribution of individuals in the population reflects the fitness landscape (evaluation function) as generation proceeds. In the proposed method, linkage groups are identified by estimating the correlation coefficient of parameter values of individuals in a population. To measure nonlinear correlation, we devised a piecewise interval correlation by iteration (PICI) algorithm. By applying SPX operator to each identified linkage group, we can preserve linkage information in recombination.

We have used two types of test functions: (1) functions which combine a tight linked sub-function and a sub-function which has no linkage among parameters; and (2) functions which consist of multiple tight linkage sub-functions with order 2. Without linkage identification, it was difficult for the algorithm with SPX to find the optimal solution on these test functions. With the proposed linkage identification method, the linkage among parameters was identified well and the algorithm found optimum solutions fairly well on those test functions.

However, proposed method did not work well on functions which consist of tight linkage sub-functions with higher order. We discussed causes of the problems involved with the proposed method and showed possible solutions to the problems, namely, linkage identification with multi-stage evolution (LIME) and Linkage identification with offline evolution (LIOF). More work is needed but we are hopeful that LIME and/or LIOF will
prove to be reliable over all the test cases herein.

Although we focused our study on linkage identification with SPX, the approach can be expected to be useful also for linkage identification with other crossover operators for real-coded GAs.

Acknowledgments

The authors would like to thank Martin Pelikan for valuable discussions and useful comments that helped to shape the paper. Professor Goldberg's contribution to the work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. Support was also provided by a grant from the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, the U. S. Army, or the U. S. Government.

References


