Two-edge Graphical Linkage Model for DSMGA-II

Author
Address
Address
Address
email

Author
Address
Address
Address
email

Author
Address
Address
Address
email

Author
Address
Address
Address
email

ABSTRACT

DSMGA-II, a model-based genetic algorithm, is capable of solving optimization problems via exploiting sub-structures of the problem. In terms of number of function evaluations (NFE), DSMGA-II has shown superior optimization ability to LT-GOMEA and hBOA on various benchmark problems as well as real-world problems. This paper proposes a two-edge graphical linkage model, which customizes recombination masks for each receiver according to its alleles, to further improve the performance of DSMGA-II. The new linkage model is more expressive than the original dependency structure matrix (DSM), providing far more possible linkage combinations than the number of solutions in the search space. To reduce unnecessary function evaluations, the two-edge model is used along with the supply bounds from the original DSM. Some new techniques are also proposed to enhance the model selection efficiency and to reduce the probability of cross-competition. Combining these proposed techniques, the empirical results show up to 30% of NFE reduction compared with the original DSMGA-II.

CCS Concepts

•Machine learning approaches → Maximum entropy modeling; Genetic algorithms; *Mixture models;*

Keywords

Genetic Algorithm; Estimation-of-Distribution Algorithm; Linkage Learning; Model Building

1. INTRODUCTION

Dependency structure matrix genetic algorithm-II (DSMGA-II) is a model building GA proposed by Hsu and Yu in 2015 [6]. Based on the dependency structure matrix (DSM), a new linkage model, called the incremental linkage set (ILS), is adopted in DSMGA-II to provide potential models for mixing. The restricted mixing and the back mixing are the major recombination operators of DSMGA-II. They are the keys to significantly reduce the number of function evaluations (NFE) compared with other optimal mixing operators. Experiment results show that DSMGA-II requires fewer function evaluations than LT-GOMEA [1] and hBOA [8], two cutting-edge evolutionary algorithms, on various benchmark problems.

However, there is still room for improvement with model building in DSMGA-II. In this paper, we propose a two-edge graphical linkage model, which is more expressive than the original DSM and customizes recombination for every single chromosome. We also demonstrate how the early-stop criterion reduces unnecessary trials and yields fewer function evaluations.

The remainder of this paper is organized as follows. Section 2 introduces the original DSMGA-II schema. Section 3 introduces the two-edge linkage model in detail, along with some new techniques to enhance mixing effectiveness. Section 4 shows the experiment results. Section 5 gives the summary and conclusion.

2. DSMGA-II

In this section, we first introduce the framework of DSMGA-II and the concept of incremental linkage set. Then we details the restricted and back mixing operators, which are the kernel operators of DSMGA-II.

2.1 Framework of DSMGA-II

DSMGA-II consists of four major components: pair-wise linkage detection, model building, restricted mixing and back mixing.

First, DSMGA-II randomly initializes a population. Then, in order to enhance the quality of pairwise linkage model and reduce the noise in the population, DSMGA-II performs a bit-flipping greedy hill climbing (GHC) on each chromosome. For each randomly initialized chromosome, GHC randomly flips each bit in the chromosome and evaluates its fitness. If the fitness improves, the chromosome accepts the change. Otherwise, the flipped bit is restored. Pairwise linkage detection was adopted in a later version of LTGA [9] and DSMGA [14] due to its resistance to sampling noise.

GECCO'17, July 15-19, 2017, Berlin, Germany

ACM ISBN TBA.

DOI: 10.1145/1235

Algorithm 1: DSMGA-II

```
$P$: population, $S$: selected population,
$s$: selection pressure, $R$: constant,
$DSM$: dependency structure matrix, $M$: mask
input  : $\ell$: problem size, $p$: population size
output: $P_{best}$
$P \leftarrow$ PopulationInitialization($\ell$, $p$)
$P \leftarrow$ GHC($P$)
while not ShouldTerminate do
    $S \leftarrow$ TournamentSelection($P$, $s$)
    $DSM \leftarrow$ UpdateMatrix($S$)
    for $k \leftarrow 1$ to $R$ do
        $I \leftarrow$ random permutation from 1 to $p$
        for $i \in I$ do
            $P_i$, $M \leftarrow$ RestrictedMixing($P_i$)
            if $M \neq \emptyset$ then
                $P \leftarrow$ BackMixing($P_i$, $M$)
return best individual in $P$
```

Performing GHC before linkage model building can further ensure the pairwise linkage information between genes, by ruling out trivial cases that can be solved without linkage information.

After initializing the population, only the selected chromosomes are used to build the linkage model. The chromosomes are chosen by a tournament selection with selection pressure as 2, suggested in [16]. DSMGA-II adopts mutual information as pairwise linkage measure and stores the linkage information in a DSM. The DSM is updated only once in each generation in order to prevent overfitting from frequent model building. Notice that tournament selection is only performed to choose chromosomes for model building instead of updating the population. Then recombinations via the restricted and back mixings are proceeded with the original population.

Before mixing, DSMGA-II builds the ILS with the linkage information stored in DSM. The ILS is a set of models indicating possible subproblem structures. In the restricted mixing, the receiver tries to flip the bits within the model. If the fitness does not decrease, the receiver becomes a donor during the back mixing, and the new pattern is tried on all chromosomes in the population. The population is randomly shuffled in advance so that each chromosome takes turns acting as the receiver during the restricted mixing and the donor during the back mixing. The pseudo code of DSMGA-II is given in Algorithm 1. The detailed implementations of the ILS and the mixing operators are described in the following sections.

2.2 Incremental Linkage Set

The DSM is an adjacent matrix representing the dependency between two variables, where each entry stores the pairwise information between two bits. In DSMGA-II, pairwise dependencies are measured by mutual information [7]. Formally, the mutual information of two random variable X and Y can be defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)},$$

where x and y are the outcomes of X and Y. In pairwise information, the random variables X and Y follows the Bernoulli distribution with support $\{0, 1\}$ and $p(x)$ represents the portion of a bit with value 1 in the population. The linkage measure can be further derived as:

$$\begin{aligned} I(X;Y) = P_{00} log \frac{P_{00}}{P_{0*}P_{*0}} + P_{11} log \frac{P_{11}}{P_{1*}P_{*1}} \\ + P_{01} log \frac{P_{01}}{P_{0*}P_{*1}} + P_{10} log \frac{P_{10}}{P_{1*}P_{*0}} \end{aligned} \quad (1)$$

The value of mutual information between two bits corresponds to the probability of two bits being in the same building-block. Therefore, DSMGA-II constructs building-blocks by clustering the DSM. Instead of using traditional hierarchical clustering algorithms proposed in [11] or the average linkage clustering technique proposed in [12], DSMGA-II adopts a specific subgraph called approximation maximum-weight connected subgraph (AMWCS) to construct the ILS by iteratively adds the bit with maximum linkage into the graph.

2.3 Restricted Mixing and Back Mixing

2.3.1 Optimal Mixing

Unlike canonical genetic algorithms that generate offsprings by recombining parental solutions, DSMGA-II extends the the idea of optimal mixing (OM) [12] with two new mixing operators: the restricted mixing and the back mixing. OM evaluates a chromosome during recombination. With the information of fitness before and after mixing, the chromosome only accepts the change if its fitness improves. Thus, a noise-free decision-making can be achieved with a much smaller population size [4]. Given overlapping building blocks, OM is also capable of solving problems with overlapping structures efficiently, since it acts like building-block wise local search.

2.3.2 Restricted Mixing

In each iteration of the restricted mixing, a receiver is randomly picked from the population, and the building-blocks are provided by a mask which is chosen from the ILS. Each mask is a set of indices that indicates which bits should be flipped together during mixing operations. All masks in the ILS must go through a *supply check* to make sure that the complement pattern of the receiver exists in the population. This way, flipping the bits is equivalent to recombining the receiver with the complement pattern. The length of the masks are also bounded to be no longer than half of the problem size, since flipping the bits with a mask longer than $\ell/2$ is equivelent to performing mixing with a smaller mask on the complement chromosome.

After *supply check*, the receiver starts with the smallest subset in the mask, and flips the bits within the subset. If the fitness does not decrease after recombination, the pattern is accepted, and the restricted mixing terminates. The receiver then becomes a donor of the new pattern for the rest of the population during the back mixing.

The idea behind the restricted mixing is building-block supply [4]. We believe all the optimal subsolution fragments should exist in the current population, which had been initialized with a sufficient population size. Therefore, given the correct building block with a proper receiver, the restricted mixing conducts optimal mixing between the receiver and the chromosome with the complementary optimal pattern. The pseudo-code of the restricted mixing is provided in Algorithm 2.

Algorithm 2: Restricted Mixing

```
$V$: candidate vertices set, $V_S$: $AMWCS$ vertices set,
$ILS$: incremental linkage set, $f$: evaluation function,
$P$: population, $\ell$: problem size,
$T$: trial solution, $M$: mask,
$R_M$: pattern of $R$ extracted by $M$,
$R_M{}'$: complement pattern of $R_M$
input  : $R$: receiver
output: $R$: reciever, $M$: mask

$V \leftarrow \{1, 2, ..., \ell\}$
$V_s \leftarrow$ {a random vertex $v \in V$}
remove $v$ from $V$
while $|V| \geq \ell/2$ do
    $ILS \leftarrow ILS \cup V_S$
    $V_S \leftarrow V_S \cup$ { $v$ with the max linkage }
    remove $v$ from $V$

for $i \leftarrow 1$ to $|ILS|$ do
    $M \leftarrow ILS_i$
    if $R_M{}' \subset P$ then
        $T \leftarrow R$
        $T_M \leftarrow R_M{}'$
        if $f(T) \geq f(R)$ and $T \notin P$ then
            $R \leftarrow T$
            return ($R$, $M$)

return ($R$, $\emptyset$)
```

2.3.3 Back Mixing

During the back mixing, all the chromosomes in the population are mixed with the pattern accepted during the restricted mixing. Chromosomes are set to accept the new pattern only with strict fitness improvement by default. However, if no fitness improves in the whole population, then the mixing which results in equal fitness is also allowed. The back mixing acceptance criterion is set differently from the restricted mixing in order to tackle real-world problems with plateaus and basins. Many operators, such as the forced improvement [1], have been developed to deal with multiple equal-quality solutions. Strict mixing improvement criterion often causes numerous evaluations to jump out of the plateaus. On the other hand, allowing all chromosomes to accept the patterns with equal fitness results in a strong drifting effect. Back mixing handles the diversity issue with a default strict-improvement criterion. When no improvement occurs, it switches to the equal-acceptance criterion to reduce unnecessary evaluations on plateaus. The empirical results suggest that the back mixing is able to deal with plateaus. The pseudo-code of the back mixing is provided in Algorithm 3.

3. THE TWO-EDGE GRAPHICAL LINKAGE MODEL

This section introduces the concept and the flow of AMWCS construction. In the original AMWCS, nodes are connected with only one linkage. We call this technique the one-edge graphical linkage model. In this section, we first describe the new linkage graph, called the two-edge graphical linkage model, which gives customized building-block models for each chromosome. Then, we discuss *supply bound*, the theoretical support behind this measure. Finally, we provide some techniques that enhance the model selection efficiency and reduce the probability of cross-competition.

Algorithm 3: Back Mixing

```
$P$: population, $f$: evaluation function,
$T$: trial solution, $E$: set of candidate solutions
input  : $D$: donor, $M$: mask
output: $P$: population

$improved \leftarrow false$
for $j \leftarrow 1$ to $|P|$ do
    $T \leftarrow P_j$
    $T_M \leftarrow D_M$
    if $f(T) \geq f(P_j)$ then
        $P_j \leftarrow T$
        $improved \leftarrow true$
    else
        if $f(T) = f(P_j)$ then
            $E \leftarrow E \cup \{T\}$

if not improved then
    accept all solutions in $E$
return $P$
```

3.1 Two-edge Graph

In the scheme of two-edge graph, the dependency measure between two bits is different from the original one-edge graph. The linkage measure equation is divided into two parts:

$$\begin{aligned} L(00 \cup 11) &= P_{00} log \frac{P_{00}}{P_{0*}P_{*0}} + P_{11} log \frac{P_{11}}{P_{1*}P_{*1}} \\ L(01 \cup 01) &= P_{01} log \frac{P_{01}}{P_{0*}P_{*1}} + P_{10} log \frac{P_{10}}{P_{1*}P_{*0}} \end{aligned} \tag{2}$$

The reason for such division is to protect the pattern within each building-blocks. For building-blocks with linkage $L(00 \cup 11)$, the pattern 00 might be flipped to pattern 11 and the pattern 11 might be flipped to pattern 00 during the restricted mixing. The resulting patterns, 00 and 11, are the complements of each other. Even though the patterns belong to different chromosomes after mixing, the original patterns are reserved in population. Same scenario happens with pattern 01 and 10.

In the two-edge graph scheme, the equation for finding a node that maximize the weight of AMWCS is defined as:

$$index = argmax \sum I(v_i; v_j), v_i \in V, v_j \in V_s, \tag{3}$$

where V is the candidate vertices set, and V_s is the AMWCS vertices set.

It is the same as the equation in the one-edge scheme. However, unlike the one-edge graph that views all patterns of a building-block equally, the two-edge linkage model takes the alleles of receivers into account during model construction. The rule of edge selection in the two-edge graph is as follows:

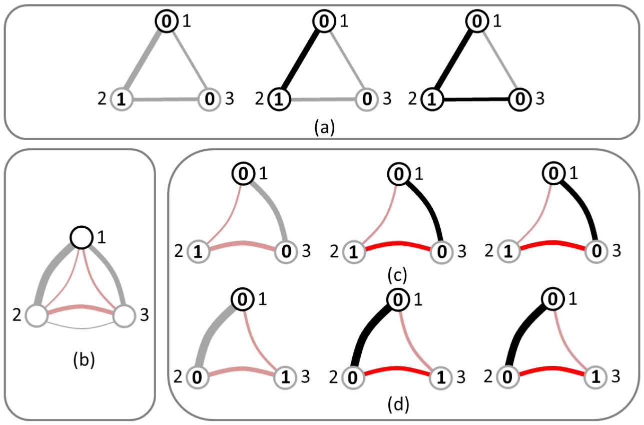
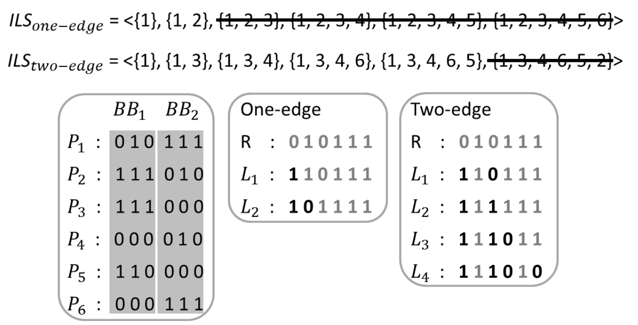


Figure 1: (a) and (b) to (d) show the construction of one-edge graph and two-edge graph, respectively. The value in each node represents the allele. The width of each edge corresponds to the strength of dependency measure between pairs. For (b) to (d), the gray and black edges represent the L(00∪11) edge, while the pink and red edges represent the L(00∪11) edge. Nodes and edges with color black or red represent the determined AMWCS.

$$L(X;Y) = \begin{cases} L(00 \cup 11), \text{ if pattern is 00 or 11} \\ L(01 \cup 10), \text{ if pattern is 01 or 10} \end{cases} \quad (4)$$

Figure 1 depicts different results of model construction with a problem of three bits, and a receiver with alleles $\{0, 1, 0\}$. First, the node 1 with allele 0 was randomly chosen from the candidate set $\{1, 2, 3\}$. For the one-edge scheme in Figure 1(a), node 2 is selected with the strongest edge $\{1, 2\}$. After two iterations, the inserted node sequence of the AMWCS is $Q = \langle\{1, 2, 3\}\rangle$, and the ILS is $\langle\{1\}, \{1, 2\}, \{1, 2, 3\}\rangle$. For the two-edge scheme in Figure 2(b), although there is a strong gray edge between nodes 1 and 2, the pattern 10 conflicts with the meaning of the gray edges L(00∪11). According to the linkage selection rule, the thin red edge $L(01 \cup 10)$ represents the linkage between nodes 1 and 2 with pattern 01. Therefore, after a receiver is determined in Figure 2(c), we remove all the conflict edges for clarity. Unlike the one-edge scheme, node 3, with the strongest edge $\{1, 3\}$, is picked in Figure 2(c). The inserted node sequence of the AMWCS is $Q = \langle\{1, 3, 2\}\rangle$, and the ILS is $\langle\{1\}, \{1, 3\}, \{1, 3, 2\}\rangle$. In Figure 2(d), a receiver with pattern 001 and a random start node 1 are used in the two-edge graph construction. When another receiver is chosen, different conflict edges are removed, resulting in a customized ILS. Although the resulting ILS $\langle\{1\}, \{1, 2\}, \{1, 2, 3\}\rangle$ is the same as one-edge graph in Figure 1(a), the two-edge graph is only constructed with non-conflict edges.

Consider a longer instance with an optimal subsolution 111 similar to the three-bit problem in Figure 1. Strong linkages can be detected among these three bits. Following the procedures described above, the one-edge graph constructs the ILS as $\langle\{1\}, \{1, 2\}, \{1, 2, 3\}\rangle$. However, None of mask in the ILS can flip the receiver with pattern 001 to 111. In short, even if one-edge graphical linkage model detects the correct model, it might still fail during mixing since the pattern might not be the optimal subsolution. On the other hand, two-edge graphical linkage model handles this problem by taking the alleles of receivers into account and preserve the correct pattern during mixing. For the two-edge scheme, choosing the mask $\{1, 3\}$ can help flipping the pattern 010 to the the optimal subsolution 111.


$ILS_{one-edge}$ = <{1}, {1, 2}, ~~{1, 2, 3}, {1, 2, 3, 4}, {1, 2, 3, 4, 5}, {1, 2, 3, 4, 5, 6}~~>

$ILS_{two-edge}$ = <{1}, {1, 3}, {1, 3, 4}, {1, 3, 4, 6}, {1, 3, 4, 6, 5}, ~~{1, 3, 4, 6, 5, 2}~~>

	BB_1	BB_2
P_1 :	010	111
P_2 :	111	010
P_3 :	111	000
P_4 :	000	010
P_5 :	110	000
P_6 :	000	111

One-edge

R :	010111
L_1 :	110111
L_2 :	101111

Two-edge

R :	010111
L_1 :	110111
L_2 :	111111
L_3 :	111011
L_4 :	111010

Figure 2: Population P is on the left, and receiver $R = P_1$ is on the right. For both building-blocks 1 and 2, the subproblem optimal pattern is 010, and the suboptimal patterns are 000, 111. Masks with no supply of complement patterns are crossed out. Restricted mixing adopting the one-edge ILS stops at L_2 due to lack of supply for longer masks. Restricted mixing adopting the two-edge ILS stops at L_4 since a chromosome with equal fitness is generated.

We believe that the ratio of a pattern in the population corresponds to the possibility of such pattern being the optimal subsolution. The two-edge model tends to align the alleles of receiver with the dominant patterns in the population. The reason for such tendency lies in the characteristic of mutual information. Since the mutual information between two bits is divided into two parts in the two-edge graph, the linkage containing the high-ratio pattern is always positive, while the other part is often negative. In the example above, if the optimal subsolution 111 is prominent in the population, then the linkages $L(00 \cup 11)$ among these nodes should be positive, and the linkages $L(01 \cup 10)$ are likely negative.

3.2 Supply bound

Although taking the receivers into account enhance the effectiveness of bit-flipping, we notice the two-edge scheme costs more evaluations in the mixing stage, especially during the back mixing. The reason behind this phenomenon is *supply overfitting*. After selecting an edge in the two-edge graph, the possible patterns for each pair of bits are narrowed down to either $\{00, 11\}$ or $\{01, 10\}$. Besides, we choose the node that gives the maximum linkage during model construction, meaning that the two selected patterns should have higher ratio in the population. Therefore, given a model constructed by a two-edge graph implies a higher possibility of finding the complement pattern in the population. The models that are customized for a specific receiver a greater chance to pass the supply check.

Figures 2 demonstrate the supply bound difference between one-edge graph and two-edge graph for a receiver 010111. The one-edge graph has shorter supply length since the complement patterns do not exist in the population. The two-edge graph is customized for the receiver, thus has a higher possibility in finding the complement patterns. Compared to the model constructed by one-edge graph, the masks in the two-edge ILS are more likely to pass the supply check.

	Supply length		Failure NFE in BM	
Problems	One-edge	Two-edge	One-edge	Two-edge
NK-S1	6.9	12.6	43.4	51.3
NK-S3	6.9	13.2	42.3	48.8
NK-S5	2.6	6.7	8.9	10.1

Table 1: Supply length and BM failure comparison (unit : K for NFE)

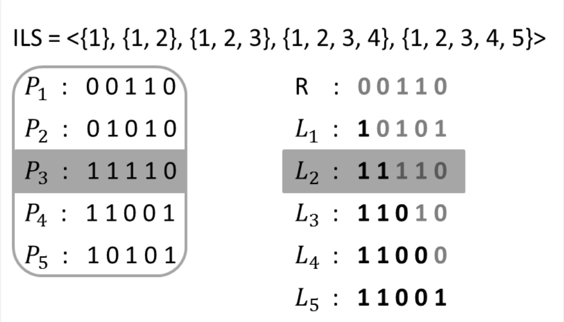
The customized model benefits the restricted mixing, but when the receiver becomes a donor during the back mixing, such customization often overfits the donor. The experiment results in Table 1 show that the average supply length of the two-edge graph is about twice longer than the one-edge counterpart. As a result, the supply bound from the two-edge graph fails to serve its original purpose: limiting the recombination trials. In addtion, the unnecessarily long patterns cause evaluation wastes and/or cross competition during the back mixing.

To prevent the supply bound from overfitting the donor, we adopt the supply length of the one-edge graph as the supply bound. The major difference is that the linkage measure in the one-edge graph is calculated with complete mutual information, which gives the global information of the population. On the other hand, the linkage measure in the two-edge graph is more skewed that favors the given receiver during model building. Therefore, using the supply bound of the one-edge graph secures the global information between bits and does not over fit a specific chromosome. This gives a better result in back mixing and reduce NFE.

Furthermore, we also find the one-edge graph supply bound benefits the restricted mixing by enhancing the success rate of recombination. We speculate that in the early generations, subproblem patterns with better fitness do not stand out from the average patterns. The dependency information in the population is rather unclear. The inaccurate models then increase the probability of failure for the restricted mixing. However, the restricted mixing operator is only terminated when the supply bound is reached of the after-mixing fitness of receiver is inferior to the original fitness. With the masks bounded by the one-edge graph supply length, unnecessary recombination trials can be reduced during the restricted mixing.

3.3 Early-Stop Criterion

We propose a new early-stop criterion for the restricted mixing that eliminates unnecessary trials and reduces NFE. During the restricted mixing, the masks in the ILS are tried in an ascending order regarding to the mask size. If the new pattern gives worse fitness value than the original one, the receiver continues to try bit-flipping with the next mask. This bit-flipping process stops only if the fitness of the generated chromosome is greater or equal to the fitness of the original receiver. However, this stop criterion might cost unnecessary evaluations since the recombination occasionally creates a chromosome that exists in the population.



ILS = <{1}, {1, 2}, {1, 2, 3}, {1, 2, 3, 4}, {1, 2, 3, 4, 5}>

P_1 : 00110	R : 00110
P_2 : 01010	L_1 : 10101
P_3 : 11110	L_2 : 11110
P_4 : 11001	L_3 : 11010
P_5 : 10101	L_4 : 11000
	L_5 : 11001

Figure 3: An example of early-stop criterion. Population P is on the left, and receiver $R = P_1$ is on the right.

```
Algorithm 4: Modified Restricted Mixing
  $ILS\_one$: one-edge incremental linkage set,
  $ILS\_two$: two-edge incremental linkage set,
  $f$: evaluation function,
  $P$: population, $\ell$: problem size,
  $T$: trial solution, $M$: mask,
  $R_M$: pattern of $R$ extracted by $M$,
  $R_M{}'$: complement pattern of $R_M$
  input : $R$: receiver
  output: $R$: reciever, $M$: mask

  $ILS\_one \leftarrow ConstructILS$(one-edge)
  $ILS\_two \leftarrow ConstructILS$(two-edge)

  for $i \leftarrow 1$ to $|ILS\_one|$ do
      $M \leftarrow ILS\_one_i$
      if $R_M{}' \not\subset P$ then
          $SupplyBound \leftarrow i$

  for $i \leftarrow 1$ to $|ILS\_two|$ do
      $M \leftarrow ILS\_two_i$
      if $R_M{}' \subset P$ and $i < SupplyBound$ then
          $T \leftarrow R$
          $T_M \leftarrow R_M{}'$
          if $T \in P$ then
              return ($R$, $\emptyset$)
          if $f(T) \geq f(R)$ then
              $R \leftarrow T$
              return ($R$, $M$)
  return ($R$, $\emptyset$)
```

Problems	Equation	Problems	Equations
Concatenated trap	$f_{m,k}^{trap}(x) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right)$ $f_k^{trap}(u) = \begin{cases} 1, & \text{if } u = k \\ \frac{k-1-u}{k}, & \text{otherwise} \end{cases}$	Folded trap	$f_{m,k=6}^{folded}(x) = \sum_{i=1}^{m} f_{k=6}^{folded}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right)$ $f_{k=6}^{folded}(u) = \begin{cases} 1, & \text{if } \lvert u = 3\rvert = 3 \\ 0.8, & \text{if } \lvert u = 3\rvert = 0 \\ 0.4, & \text{if } \lvert u = 3\rvert = 1 \\ 0, & \text{if } \lvert u = 3\rvert = 2 \end{cases}$
Cyclic trap	$f_{m,k}^{cyclic}(x) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot(k-1)-k+2}^{i\cdot(k-1)+1} x_j\right)$ $f_k^{trap}(u) = \begin{cases} 1, & \text{if } u = k \\ \frac{k-1-u}{k}, & \text{otherwise} \end{cases}$	NK-landscape	$f_{\ell,k,s}^{NK}(x) = \sum_{i=0}^{(\ell-k-1)/s}$ $f_{k,i}^{subNK}(x_{i\cdot s+1}, x_{i\cdot s+2}, ...x_{i\cdot s+k+1})$
2D Spin-glass	$f_n^{spin}(x) = -\sum_{i,j=0}^{n} x_i x_j J_{ij}$	MAX-SAT	$F = \bigwedge_{i=1}^{m}\left(\bigvee_{j=1}^{k_i} \ell_{ij}\right)$

Table 2: Equations of the bechmark problems

For example, consider a five-bit trap problem with global optimum 11111 and local optimum 00000 in Figure 3. P_1 is chosen as receiver in the restricted mixing. Suppose the masks in ILS follow the sequence $\{1, 2, 3, 4, 5\}$ and the complement patterns for all five models exist in the population. Restricted mixing failed with mask $\{1\}, \{1, 2\}$ since the patterns are inferior to the original pattern 00110 in trap problems. After performing bit-flipping with the second mask $\{1, 2\}$, the resulted pattern 11110 exists in the chromosome P_3. The original stop criterion continues the procedure, ending up with three more more failure recombinations, L_3, L_4, and L_5. These attemps are unnecessary, since now they can be performed on the existing chromosome P_3. Also, such unnecessary attempts may cause other chromosomes similar to P_3 and undesiredly reduce the diversity of the population.

To prevent from this situation, we terminate the restricted mixing whenever the new generated chromosome already exists in the population. Another receiver along with another set of customized models are chosen for the next iteration of the restricted mixing. This criterion preserves diversity in the population and also reduces NFE waste. The pseudocode of the modified restricted mixing is given in Algorithm 4.

4. EXPERIMENT RESULTS

In this section, we first briefly introduce the six benchmark problems for the following experiments. Then, the experiment setup is detailed. Finally, we compare and contrast the experiment results of the original DSMGA-II, the modified DSMGA-II, LT-GOMEA and hBOA.

4.1 Test Problems

Six types of linkage benchmark problems are used in this paper, including four classical linkage-underlying problems and two real-world problems. These benchmark problems each covers different aspects and characteristics of real-world problems. The *Concatenated Trap* is composed of m additively separable trap functions, each with k variables [3]. It is well known that in order to solve trap problems, the underlying structure must be detected and preserved during mixing [13]. The *Cyclic Trap* consists of overlapping trap functions with wraparound [15]. The *Folded trap* is one of the multiple variants of *NK-landscape* problems described in [5]. We use the bipolar deceptive function with the subsolution size $k = 6$. The performance depends heavily on the ability of reducing unnecessary exploration of plateaus due to the symmetric characteristics of the trap function. The *NK-landscape* functions are composed of overlapped and randomly generated sub-functions [10]. Each instance is controlled by three parameters: the problem size ℓ, the number of neighbors of one gene k, and the step size s, i.e. the offset of two adjacent sub-functions. We use *NK-landscape* with step 1, 3 and 5 to represent problems with different degrees of overlapping in our experiments. The *Ising spin-glass* gives a set of variables in one of the two states {+1, -1}. For each pair of neighboring spins i and j, there exists a coupling constant J_{ij}. The goal is to find a combination of states that satisfies the most coupling pairs. The *Maximum Satisfiability problem* (MAX-SAT) consists of a series of logical and clauses. Each clause contains a series of logical or variables. It is a classical NP-complete problem. For our experiments, we use the Uniform Random-3-SAT instances from SATLIB[1] with all satisfiable clauses.

4.2 Experiment Setup

DSMGA-II is an enhanced edition of DSMGA and it outperforms multiple variants of its predecesors [14] in all benchmark problems. Thus, we will not discuss DSMGA in the following comparison. DSMGA-II is based on the idea of OM, so it is necessary to compare DSMGA-II with GOMEAs. In our experiments, we use the version of LT-GOMEA with forced improvement [2]. Although several different linkage models have been proposed over the years [2], LT-GOMEA [1] is still considered a state-of-the-art OM algorithm. Also, we compare DSMGA-II with the hierarchical Bayesian optimization algorithm (hBOA) [8] Similar to LT-GOMEA, hBOA is also a milestone and is often used in recent Model-Building GA researches.

[1] http://www.satlib.org

[2] http://homepages.cwi.nl/∼bosman/source_code.php

A minimum population is required for a certain number of consecutive successful runs. However, the conventional bisection procedure [8] often fails to yield the minimum NFE since the minimum population size varies in the bisection procedure for different algorithms. In the following experiments, we adopt an adaptive sweeping procedure described in [6] for more accurate search of minimum NFE. The sweeping procedure gives better precision than the canonical bisection procedure, since the acquired resolution increases as the steps decreases after each successful trial. The minimum evaluations required for each problem are averaged over 100 independent runs or instances.

We compare DSMGA-II, LT-GOMEA, and hBOA under the following settings. The selection pressure is set as 2 for all three algorithms. For DSMGA-II, the model building is performed every $\ell/50$ generations. LT-GOMEA is performed without local searh to enhance its performance, as in [1].

4.3 Results

4.3.1 Comparison with the original DSMGA-II

Problems	Original	Improved	Ratio
Cyclic trap, $\ell = 400$	123.1	111.5	9.4%
Folded trap, $\ell = 480$	354.5	243.2	31.4%
NK-S1, $\ell = 400$	767.4	701.6	8.5%
NK-S3, $\ell = 400$	550.5	533.5	3.1%
NK-S5, $\ell = 400$	62.0	59.5	4.1%
2D spin-glass, $\ell = 784$	863.2	684.5	20.7%
MAX-SAT, $\ell = 200$	7593.3	5655.1	25.5%

Table 3: Required NFE of DSMGA-II for the largest test problems (unit: K)

Compared with the original DSMGA-II, the modified DSMGA-II with our new techniques show superior performance on all six benchmark problems. In Table 3, comparison between our new techniques along with the original DSMGA-II are shown only with the largest problem size since the improvement is relatively minor with smaller problems. The results indicate that the modified DSMGA-II with our new techniques can reduce NFE up to 30% compared to the original DSMGA-II.

Our new technique is most effective when dealing with the folded trap problem. The two-edge graph is capable of constructing customized models for a chosen receiver. This characteristic shows great advantage when facing problems with multiple equal-quality subsolutions. Since the two-edge model tends to align with the most promenant pattern in the population, it costs fewer evaluations to jump out of plateaus by converging through stronger drifting effect.

This characteristic is also helpful when solving the Ising spin-glass problems. There are often two equal-fitness subsolutions in the spin-glass problem. The original DSMGA-II can only detect the strong linkage of the subsolution without knowing the correct pattern of the subsolution to contribute to global fitness. This results in a significant waste of function evaluations on repeatedly flipping subsolution patterns to another equally fit subsolution. With the two-edge graph, different models can be generated from the same graph. Empirically, this helps preserve the optimal patterns during mixing and reduce attempts of trying the complementary pattern.

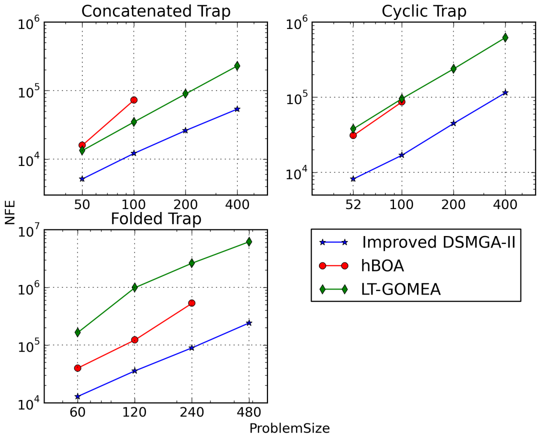


Figure 4: Scalability of the modified DSMGA-II, LT-GOMEA and hBOA on the problems of deceptive variants.

The modified DSMGA-II requires fewer evaluations than both LT-GOMEA and hBOA on all six benchmark problems. It is worth mentioning that most of the slope of DSMGA-II data line decreases as the problem size increases. This indicates that the modified DSMGA-II excels the original one in scalability. The following comparisons are divided into three categories according to different characteristics of the problems.

4.3.2 Comparison on deceptive variants

We first consider the deceptive problems in Figure 4. For concatenated trap problems, the modified DSMGA-II requires fewer evaluations than LT-GOMEA and hBOA. For folded trap problems, the two-edge model helps aligned the receiver with the promenant pattern in population. Together with the back mixing, the pattern soon dominates the population, and the population is able to jump out of the plateau. This strong and precise drifting effects is the key to the outstanding performance of DSMGA-II on folded trap problems.

For cyclic trap, most of patterns converges to three, four or five consecutive 1s or 0s after local search. Afterwards, it is crutial to construct a model that contains all zeros in order to be flipped to all ones. The original graph only detects the subproblem structures. However, the two-edge graph further constructs a most suitable structure for a specific receiver.

4.3.3 Comparison on NK-landscape problems

The results of the NK series in Figure 5 demonstrate the ability of the ILS model in handling randomly generates overlapping problems. The difference between the modified DSMGA-II and other algorithms enlarges as the degree of overlapping decreases. When coping with severly overlapping structures, DSMGA-II requires more evaluations during graph refining process. Once the linkage information stands out, the back mixing operator allows the correct pattern to dominate the others in the population instantly. The two-edge model provides far more linkage combinations than

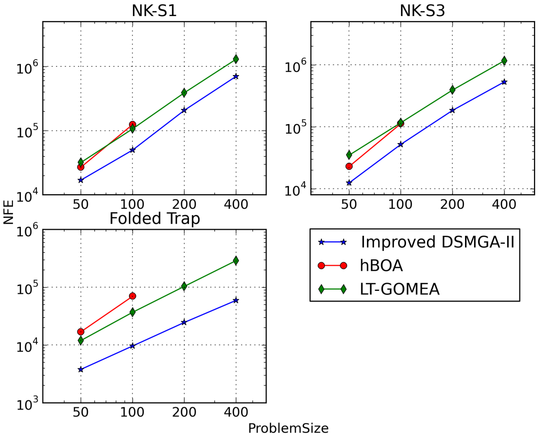


Figure 5: Scalability of the modified DSMGA-II, LT-GOMEA and hBOA on the NK-landscape problems with various degrees of overlapping.

the number of solutions in the search space. Generating ILS from the abundant combinations in the two-edge graph results in more flexible and better customized models. These models enable DSMGA-II to handle overlapping problems nicely.

4.3.4 Comparison on real-world problems

For results of Ising spin-glass problems in Figure 6, the slope of DSMGA-II decreases as the problem size increases. This is due to the numerous numbers of plateaus in the spin-glass problems. As discussed in Section 4.3.1, converging the population for plateaus is where the advantages of DSMGA-II holds. The MAX-SAT problems, while the NFE grows exponentially, which is reasonable since MAX-SAT is known to be NP-hard, the modified DSMGA-II still requires the fewest evalutions among the three algorithms.

5. CONCLUSIONS

In this paper, we proposed a two-edge graphical linkage model for DSMGA-II. The new model is more expressive than the original one and provides customized recombination masks for every single chromosome. To reduce unnecessary function evaluations, a new criterion of supply check is also designed. Combined with the other minor modification, the early stop, the new linkage model shows a stable scalability and improves the original DSMGA-II by up to 30% in terms of NFE.

Researches concerning model-building techniques, providing centralized information for recombination, have improved the canonical genetic algorithms by much. We find it interesting that distribution of such centralized techniques (customization the recombination for each chromosome) again can improve the performance of model-building genetic algorithms. Such improvement may be applied to other evolutionary algorithms as well; of course, further investigations, especially from the theoretical aspect, are needed.

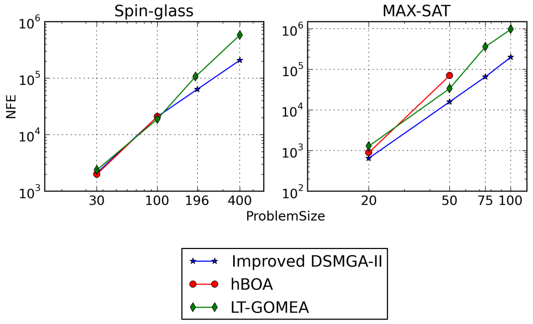


Figure 6: Scalability of the modified DSMGA-II, LT-GOMEA and hBOA on Spin-glass and MAX-SAT (*LT-GOMEA fails to reach the global optima for 2 out of 100 instances on 100-bit MAX-SAT).

6. REFERENCES

[1] P. A. Bosman and D. Thierens. Linkage neighbors, optimal mixing and forced improvements in genetic algorithms. In *Proceedings of the 14th annual conference on Genetic and evolutionary computation*, pages 585–592. ACM, 2012.

[2] P. A. Bosman and D. Thierens. More concise and robust linkage learning by filtering and combining linkage hierarchies. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 359–366. ACM, 2013.

[3] K. Deb and D. E. Goldberg. Sufficient conditions for deceptive and easy binary functions. *Annals of mathematics and Artificial Intelligence*, 10(4):385–408, 1994.

[4] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *COMPLEX SYSTEMS*, 6:333–333, 1992.

[5] D. E. Goldberg, K. Deb, and J. Horn. Massive multimodality, deception, and genetic algorithms. *Urbana*, 51:61801, 1992.

[6] S.-H. Hsu and T.-L. Yu. Optimization by pairwise linkage detection, incremental linkage set, and restricted/back mixing: DSMGA-II. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, pages 519–526. ACM, 2015.

[7] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[8] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Genetic and Evolutionary Computation Conference*, pages 1271–1282. Springer, 2003.

[9] M. Pelikan, M. W. Hauschild, and D. Thierens. Pairwise and problem-specific distance metrics in the linkage tree genetic algorithm. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 1005–1012. ACM, 2011.

[10] M. Pelikan, K. Sastry, D. E. Goldberg, M. V. Butz, and M. Hauschild. Performance of evolutionary algorithms on NK landscapes with nearest neighbor interactions and tunable overlap. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pages 851–858. ACM, 2009.

[11] D. Thierens. The linkage tree genetic algorithm. In *International Conference on Parallel Problem Solving from Nature*, pages 264–273. Springer, 2010.

[12] D. Thierens and P. A. Bosman. Optimal mixing evolutionary algorithms. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 617–624. ACM, 2011.

[13] D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. *Urbana*, 51:61801, 1993.

[14] T.-L. Yu, D. E. Goldberg, A. Yassine, and Y.-p. Chen. Genetic algorithm design inspired by organizational theory: Pilot study of a dependency structure matrix driven genetic algorithm. In *Genetic and Evolutionary Computation Conference*, pages 1620–1621. Springer, 2003.

[15] T.-L. Yu, K. Sastry, and D. E. Goldberg. Linkage learning, overlapping building blocks, and systematic strategy for scalable recombination. In *Proceedings of the 7th annual conference on Genetic and evolutionary computation*, pages 1217–1224. ACM, 2005.

[16] T.-L. Yu, K. Sastry, D. E. Goldberg, and M. Pelikan. Population sizing for entropy-based model building in discrete estimation of distribution algorithms. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 601–608. ACM, 2007.