A New Technique 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 20% 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 novel model building GA proposed by Hsu and Yu in 2015 [4]. 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. Restricted mixing and back mixing are the two major 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 less NFE than some cutting-edge evolutionary algorithms, such as LT-GOMEA [1] and hBOA [6], on various benchmark problems.

However, there is still room for improvement with model building in DSMGA-II. In this paper, we consider building a two-edged approximation maximum-weight connected subgraph (AMWCS) as a more effective linkage model. We also demonstrate how early-stopping methods enable more efficient model selection during mixing, resulting in less NFE.

This paper is organized in six parts. Section 2 introduces the original DSMGA-II schema. Section 3 introduces the 2-edged linkage model in detail, along with some new techniques to enhance mixing effectiveness. Section 4 shows the experiment results and 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. After that, we give details of restricted mixing and back mixing, which are the kernel operators of DSMGA-II.

2.1 Framework of DSMGA-II

DSMGA-II consists of four major steps: population initialization, linkage 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 [7] and DSMGA [11] due to its resistance to sampling noise. Performing GHC before linkage model building can further ensure the pairwise linkage information between genes, by

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 : l: problem size, p: population size
output: P_{best}

```
P ← PopulationInitialization(l, p)
P ← GHC(P)
while not ShouldTerminate do
    S ← TournamentSelection(P, s)
    DSM ← UpdateMatrix(S)
    for k ← 1 to R do
        I ← random permutation from 1 to p
        for i ∈ I do
            P_i, M ← RestrictedMixing(P_i)
            if M ≠ ∅ then
                P ← BackMixing(P_i, M)
return best instance in P
```

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 [13]. 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. The following steps of restricted mixing and back mixing proceed 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 for restricted mixing. In restricted mixing, the receiver tries to flip the bits within the model. If the fitness does not decrease, the receiver becomes the donor in back mixing and the new pattern is tried on all chromosomes in the population. The population is randomly shuffled before restricted mixing so that each chromosome takes turns acting as the receiver in restricted mixing and the donor in 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 [5]. 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 (binomial) Bernoulli distribution with support {0, 1}. p(x) represents the portion of a bit having value 1 in the population. Therefore, the linkage measure can be further derived as:

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

If X and Y are independent, then I(X; Y) equals to 0. Otherwise, the mutual information indicates how much the corresponding bits differ from most of the chromosomes in current population. 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 [9] or the average linkage clustering technique proposed in [10], DSMGA-II adopts a specific subgraph called approximation maximum-weight connected subgraph (AMWCS). DSMGA-II constructs AMWCS from a certain bit, and iteratively adds one single bit into the graph.

2.3 Restricted Mixing and Back Mixing

2.3.1 *Optimal Mixing*

Unlike canonical genetic algorithms that generate offsprings by recombining parent solutions, DSMGA-II extends the the idea of optimal mixing (OM) [10] with two new mixing operators: restricted mixing and 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 [2]. 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 in each iteration of 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 indexes 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. 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 restricted mixing terminates. The receiver then becomes the donor of the new pattern for rest the population in back mixing.

The idea behind restricted mixing is building-block supply [2]. 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, restricted mixing conducts optimal mixing between the receiver and the chromosome with the complementary optimal pattern. The pseudo-code for 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, l: 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

$V \leftarrow \{1, 2, ..., l\}$
$V_s \leftarrow$ a random vertex $v \in V$
remove v from V
while $|V| \geq l/2$ **do**
  $ILS \leftarrow ILS \cup V_S$
  $V_S \leftarrow V_S \cup \{$ the nearest vertex $v \in V\}$
  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

In back mixing, every chromosome in the population is mixed with the pattern accepted in restricted mixing. Chromosomes are set to accept the new pattern only with strict fitness improvement by default. However, if no chromosome fitness improves in the whole population, then the mixing which results in equal fitness is also allowed. The back mixing acceptance criteria is set differently from 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 criteria 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 criteria. When no improvement occurs, it switch to equal-acceptance criteria to reduce unnecessary evaluations on plateaus. The empirical experiment results suggest that back mixing is able to deal with both plateaus and diversity issues. The pseudo-code for restricted mixing is provided in Algorithm 3.

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. THE NEW 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 AMWCS. In this section, we first describe the new linkage graph, called the two-edge AMWCS, 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.

3.1 2-edge AMWCS

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

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

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 11 and the pattern 11 might be flipped to 00 during restricted mixing. The resulting patterns, 00 and 11, are the complements of each other, and the original patterns are reserved during mixing. Same scenario happens with pattern 01 and 10.

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

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

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 AMWCS that view all patterns of a building-block equally, the alleles of receivers are taken into account in the two-edge linkage model. The rule of edge selection in the two-edge AMWCS is as follows:

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

Figure 2 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

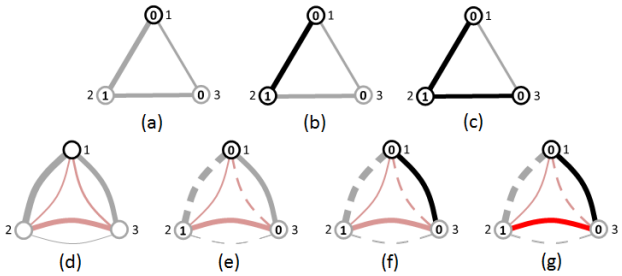

Figure 1: (a) to (c) and (d) to (g) show the construction of one-edge AMWCS and two-edge AMWCS, respectively. The value in each node represents the allele. The width of each edge corresponds to the strength of dependency measure between pairs. For (d) to (g), the gray and black edges represent the L(00∪11) 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.

Figure 2(b), node 3 is selected with the strongest edge {1, 2}. After two iterations, the inserted node sequence of the one-edge AMWCS is $Q = \langle\{1, 2, 3\}\rangle$, and the incremental linkage set is ILS = $\langle\{1\}; \{1, 2\}; \{1, 2, 3\}\rangle$. For the two-edge scheme in Figure 2(d), although there is a strong gray edge between node 1 and 3, the pattern 10 conflicts with the meaning of gray edges. According to the linkage selection rule, the thin red edge $L(01 \cup 10)$ represents the linkage between node 1 and 3 with pattern 01. For clarity, we illustrate conflict edges with dotted lines. Same scenario happens to other edges in Figure 2(e). Unlike the one-edge scheme, node 3, with the strongest solid edge {1, 3}, is picked in Figure 2(f). Therefore, the inserted node sequence of the two-edge AMWCS is $Q = \langle\{1, 3, 2\}\rangle$, and the incremental linkage set is ILS = $\langle\{1\}; \{1, 3\}; \{1, 3, 2\}\rangle$.

Consider an instance with optimal subsolution 111 with strong linkage among these three bits. Following the procedures described above, the one-edge AMWCS construct 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 AMWCS 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 AMWCS can handle this problem by taking the alleles of receivers into account and preserve the correct pattern during mixing. 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 AMWCS, the linkage containing the high-ratio pattern is often positive, while the other part is usually 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. In Figure 2(d), the two-edge AMWCS starts with node 1, and the receiver has pattern 001. Unlike the one-edge model, the ILS $\langle\{1\}; \{1, 3\}; \{1, 3, 2\}\rangle$ is only constructed with solid edges. For the two-edge scheme, choosing the mask {1, 3} can help flipping the pattern 001 to the the optimal subsolution 111.

Parameters	Continue	Break
Success rate	1,000	1000
Cross competition occurrence	1,000	1000
NFE mean	1,000	1000
NFE variance	1,000	1000

Table 1: Success rate comparison

3.2 Supply bound

Although two-edge AMWCS takes the receiver into account and performs more effect in bit flipping, we find that two-edge AMWCS leads to much more NFE wasted in the mixing stage. In back mixing, the failing rate gets higher due to the problem of supply overfitting. The reason is that the bit-pattern grouped together are complement to the other, so if the AMWCS add a node into the candidate set, it means this pattern and the complement pattern between them account for higher ratio of the population, and it leads higher probability to pass the supply check. For example, consider a receiver with allele 0, 1, 0, and we start with allele 0. If the procedure picks node 2 instead of node 3 in this iteration, this implies that the ratio of 01 and 10 pattern between node 1 and 2 is higher than the 00 and 11 pattern between node 1 and 3, and it leads easier to pass the supply check since the complement pattern has a high ratio in the population. Experiment (Table 2) shows that the length of two-edge AMWCSâĂŹs supply is about two times longer than one-edge AMWCSâĂŹs supply. If the procedure uses the overfitting supply for the back mixing, it is just like to copy the whole chromosome instead of collecting the suboptima patterns. So it makes back mixing performs less efficiently and brings about more wasted NFE in back mixing.

To avoid supply overfitting, we adopt one-edge AMWCS for the supply bound. The concept is that one-edge uses the complete mutual information instead of divided one. It means that unlike two-edge AMWCS, one-edge AMWCS follows the global information between bits and does not overfit supply. We also find that the one-edge supply bound benefits the restricted mixing. The reason we speculated is that in the early stage, the information for model building is not clear and not accurate yet, so the restricted mixing usually fails. Due to the much longer length of supply and the restricted mixing operator is terminated only if the receiverâĂŹs fitness gets better or the supply check does not pass, two-edge AMWCS wastes much more NFE in restricted mixing. With the smaller supply length one-edge AMWCS produced, it unexpected benefits the restricted mixing as well.

3.3 Chromosome Existence Check

In restricted mixing, the procedure is not terminated when the receiver is flipped to the pattern which has already existed in the population even though the flipped pattern leads to improvement. The procedure calls the continue operator instead. Nevertheless, we find that this criterion leads to extra cross-competition. For example, consider a problem which contains two subproblem with the same suboptimum

pattern 111 and pattern 000000 exists in the population. In restricted mixing, if the receiverâĂŹs pattern is 111000, even though the procedure flipped it to 000000 which has already exists, restricted mixing calls the continue operator and keeps trying bit-flipping. Thus, the chromosome can be flipped to 000111 and starts the backmixing. Under this circumstance, the chromosomes in the population with the pattern 111 in the first subproblem are possibly being replaced with pattern 000. This phenomenon increases the probability of ruining optimum and NFEs.

To prevent this situation, we substitute the continue operator to break for chromosome existence check. This method would terminate restricted mixing and change another chromosome into receiver. The pseudo code of modified restricted mixing is given in Algorithm 4.

Algorithm 4: Restricted Mixing

V: candidate vertices set, V_S: $AMWCS$ vertices set,
ILS: incremental linkage set, f: evaluation function,
P: population, l: 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 ← {1, 2, ..., l}
V_s ← a random vertex v ∈ V
remove v from V
while |V| ≥ l/2 do
    ILS ← ILS ∪ V_S
    V_S ← V_S ∪ { the nearest vertex v ∈ V}
    remove v from V
for i ← 1 to |ILS| do
    M ← ILS_i
    if R_M' ⊂ P then
        T ← R
        T_M ← R_M'
        if T ∈ P then
            return (R, ∅)
        if f(T) ≥ f(R) then
            R ← T
            return (R, M)
return (R, ∅)
```

4. EXPERIMENT RESULTS

In this section, we first describe the six benchmark problems. After that, the experiment setup is given. Finally, we illustrate the experiment results of the original DSMGA-II, the improved 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. Detail descriptions of the six benchmark problems are as follows.

4.1.1 Concatenated trap

The concatenated trap is composed of m additively separable trap function, 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 [8]. The fitness function is described as follows:

$$f_{m,k}^{trap}(x) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right)$$

where

$$f_k^{trap}(u) = \begin{cases} 1, & \text{if } u = k \\ \frac{k-1-u}{k}, & \text{otherwise} \end{cases}$$

4.1.2 Cyclic trap

The cyclic trap consists of overlapping trap functions with wraparound [12]. The fitness function is described as follows:

$$f_{m,k}^{cyclic}(x) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot(k-1)-k+2}^{i\cdot(k-1)+1} x_j\right)$$

where

$$f_{m,k}^{cyclic}(x) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot(k-1)-k+2}^{i\cdot(k-1)+1} x_j\right)$$

and

$$x_j = x_{j-l}, \text{if } l < j \leq 2l$$

Given a 12-bit cyclic trap with the size of subfunction $k = 5$, the fitness of identifying one correct subsolution, e.g. 111110000000 with $fitness = 2.2$, is lower than having all incorrect subsolutions, e.g. 000000000000 with $fitness = 2.4$. Therefore, the linkage model must not only identify the structure of subsolutions, but also take different scenarios of subsolution combinations into account for recombination to succeed.

4.1.3 Folded trap

We use the bipolar deceptive function with the subsolution size k = 6. Each subsolution contains two global optima and lots of local optima. It is one of the multiple variants of NK-landscape problems described in [3]. The performance depends heavily on the ability to reduce unnecessary exploration of plateaus, due to the symmetric characteristics of the trap function. The fitness function is described as follows:

$$f_{m,k=6}^{folded}(x) = \sum_{i=1}^{m} f_{k=6}^{folded}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right)$$

where

$$f_{k=6}^{folded}(u) = \begin{cases} 1, & \text{if } |u=3| = 3 \\ 0.8, & \text{if } |u=3| = 0 \\ 0.4, & \text{if } |u=3| = 1 \\ 0, & \text{if } |u=3| = 2 \end{cases}$$

4.1.4 NK-landscape

The NK-landscape functions are composed of overlapped, randomly generated sub-functions [8]. There are three parameters controlling the fitness function: l is the problem size, k is the number of neighbors of one gene, and so is the step size, i.e. the offset of two adjacent sub-functions. The fitness function is given as follows:

$$f_{l,k,s}^{NK}(x) = \sum_{i=0}^{(l-k-1)/s} f_{k,i}^{subNK}(x_{i\cdot s+1}, x_{i\cdot s+2}, ...x_{i\cdot s+k+1})$$

4.1.5 Ising spin-glass

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 Jij. The goal is to find a combination of states that minimizes the fitness function described as follows:

$$f_n^{spin}(x) = -\sum_{i,j=0}^{n} x_i x_j J_{ij}$$

4.1.6 MAX-SAT

The Maximum Satisfiability problem (MAX-SAT) consists of a series of *logicaland* clauses. Each clause contains a series of *logicalor* variables. It is a classical NP-complete problem. The fitness function is described as follows:

$$F = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k_i} l_{ij} \right)$$

where m is the number of clauses, k_i is the number of literals in the i-th clause and l_{ij} is the j-th literal in the i-th clause. Here, we use the Uniform Random-3-SAT instances from SATLIB [1] with all satisfiable clauses.

4.2 Experiment Setup

Since DSMGA-II is based on the idea of OM, it is necessary to compare DSMGA-II with GOMEAs.

Describe sweep, 10 hits and the average of 100 instances.

4.3 Results

Problems	Original	Improved	Ratio
Concatenated trap,l=400	1,000	1000	1%
Cyclic trap, l = 400	1,000	1000	1%
Folded trap, l = 480	1,000	1000	1%
NK-S1, l = 400	1,000	1000	1%
NK-S3, l = 400	1,000	1000	1%
NK-S5, l = 400	1,000	1000	1%
Ising spin-glass, l = 400	1,000	1000	1%
MAX-SAT, l = 200	1,000	1000	1%

Table 2: Required NFE of DSMGA-II for the largest test problems

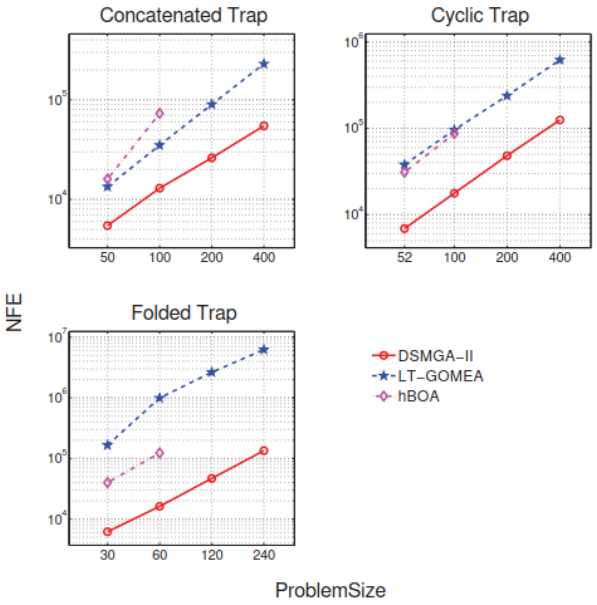

Figure 2: Scalability of DSMGA-II, LT-GOMEA and hBOA on the problems of deceptive variants.

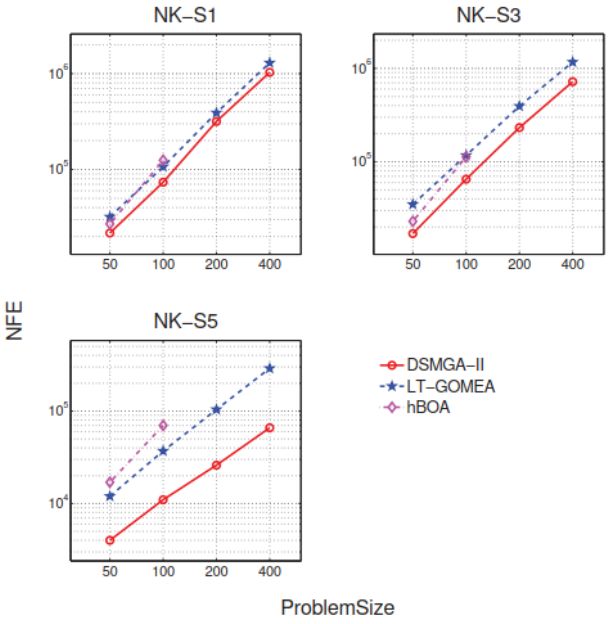

Figure 3: Scalability of DSMGA-II, LT-GOMEA and hBOA on NK-landscape problems with various degrees of overlapping.

5. CONCLUSIONS

[1] http://homepages.cwi.nl/~bosman/source_code.php

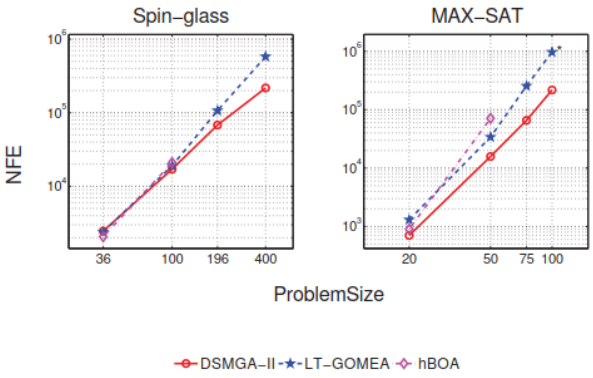

Figure 4: Scalability of DSMGA-II, LT-GOMEA and hBOA on Spin-glass and MAX-SAT (*LT-GOMEA fails to reach the global optima for two instances with $l = 100$ on MAX-SAT).

Our new techniques constructs a more precise linkage model that gives building-blocks customized for each receiving chromosome during mixing. We also show that early-stopping technique can reduce the probability of cross-competition, allowing DSMGA-II to solve the benchmark problems with a smaller population and less NFE. The improved DSMGA-II decrease NFE by up to 20% compared with the original version. It also outperforms LT-GOMEA and hBOA on multiple benchmark problems.

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

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

[4] 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.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.