國立臺灣大學電機資訊學院電機工程學系
碩士論文
Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University
Master Thesis

以雙邊圖鏈結模型改良
相依性結構矩陣遺傳演算法二代
Improving DSMGA-II
with Two-edge Graphical Linkage Model

陳品霖
Ping-Lin Chen

指導教授：于天立博士
Advisor: Tian-Yu Li, Ph.D.

中華民國 106 年 7 月
July, 2017

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

國立臺灣大學碩士學位論文
口試委員會審定書

以雙邊圖鏈結模型改良
相依性結構矩陣遺傳演算法二代

Improving DSMGA-II with
Two-edge Graphical Linkage Model

本論文係陳品霖君（學號 R04921043）在國立臺灣大學電機工程學系完成之碩士學位論文，於民國 106 年 07 月 27 日承下列考試委員審查通過及口試及格，特此證明。

口試委員： 于天立 （簽名）
（指導教授）

雷欽隆

張時中

系主任 劉志文 （簽名）

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

誌謝

首先，我要感謝我的指導老師于天立教授這兩年來的指導，不論是在學術上技術的指導，抑或者是對於做學問的態度上我都獲益良多。其中讓我感受最深刻的, 主要就是教授對於研究的執著，研究中最重要的是背後的原因，而不是在於結果。儘管結果出色，要是不明白其中道理，也是徒然。相反的，即使成果並不顯著，但假如能試著理解其中的原因，那才是有收穫的事情。除次之外，「不敢失敗，就不敢成功」也讓我銘記在心，它提醒了我，不論是研究還是人生，都要勇於嘗試。不畏懼失敗，才有機會在嘗試中獲得成功。

一路走來，首先要感謝實驗室的夥伴們，昶毅、家華、麒元、俊人、松錡、政穎、遠任、浩倫，感謝你們陪我走過兩年的研究生活。最後，當然要感謝我的家人，謝謝你們總是給我支持與鼓勵，讓我能順利的走完這段路。

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

摘要

相依性結構矩陣遺傳演算法二代是一個模型建構的基因演算法，它能夠利用探索問題的子結構來解出許多的最佳化問題。此演算法在許多指標性問題，其中包含兩個現實問題上，已經顯示了比起兩個具有代表性的演算法表現得更加卓越。在這篇論文之中，我提出了一個客製化的模型稱作雙邊圖鏈結模型，這個模型針對了每一個接收者客製化了相對應的遮罩，使得相依性結構矩陣遺傳演算法二代擁有進一步的性能提升。這個新的模型比起原來的版本，提供了夠多種可能的重組方式。為了減少一些不必要的嘗試，這個模型和供給限制的技術一起搭配使用。此外，這篇論文也提供了一個新的技術稱作提早終止判定，些微地增進了重組的效率。結合以上提出的技術，性能在指標性問題的結果上比起原版的相依性結構矩陣遺傳演算法二代有了平均十二個百分點的提升。

關鍵詞: 演化式演算法、基因演算法、鏈結學習、模型建構。

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Abstract

DSMGA-II, a model building genetic algorithm, is able to solve optimization problems via exploiting sub-structures of the problem. DSMGA-II has shown superior optimization ability to LT-GOMEA and hBOA on several benchmark problems including two real-world problems, Ising spin-glass and MAX-SAT. In this thesis, I propose a customized model called two-edge graphical linkage model, which customizes the recombination masks for each receiver according to its alleles, to further improve the performance of DSMGA-II. The new linkage model provides far more possible linkage combinations than the original version. To reduce unnecessary trails, the two-edge model is combined with the supply bounds from the original model. A new techniques called early stop criterion is also proposed to slightly enhance the efficiency in mixing. Combining these proposed techniques, the empirical results show an average of 12% NFE reduction on the benchmark problems compared with the original DSMGA-II.

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Contents

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

List of Figures

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

List of Tables

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Chapter 1

Introduction

Background

Evolutionary algorithms (EAs) are commonly used for solving optimization problems [12]. Inspired by biological evolution, EAs keep a population and use mechanisms such as reproduction, mutation, recombination, and selection. Candidate solutions play the role of individuals in a population, and the fitness function determines the quality of the solutions. The population is evolved toward better solutions after the repeated application of the above operators (Figure 1.1).

Genetic algorithm (GA) is one of the types of EAs. The operators of GA include selection, crossover (sometimes called mixing), mutation and replacement [6,12] (Figure 1.2(a)). In GAs, for easy and small-scale problems, simple heuristics works quite well to find the global or near-global optima for the given fitness function. However, to solve the more difficult or large-scale problems, the traditional crossover might disrupt the hidden orders or the structures of the problems (Figure 1.3). Thus, problem decomposition is the key to solving problems more effectively and efficiently. The highly related variables in the problem should be treated together in the crossover phase. The relationship between the variables which should be considered jointly to avoid disruptions is called linkage. Since the importance of the linkage and the problem decomposition has been addressed in the GA field [5,11], the model building genetic algorithm (MBGA) [15,18,20] which adopts the concept of linkage learning have been developed (Figure 1.2 (b)).

doi:10.6342/NTU201704045

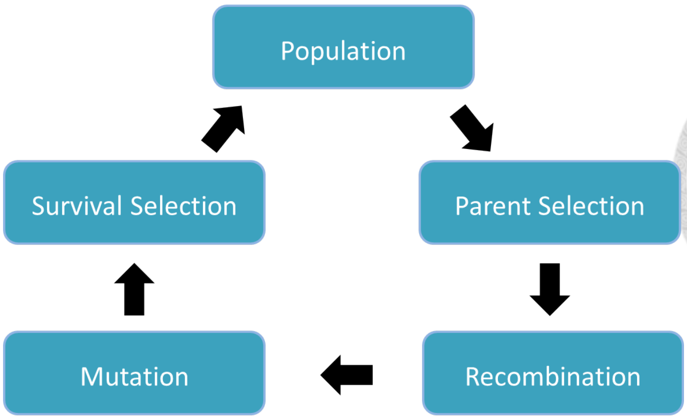


Figure 1.1: The mechanisms of EA.

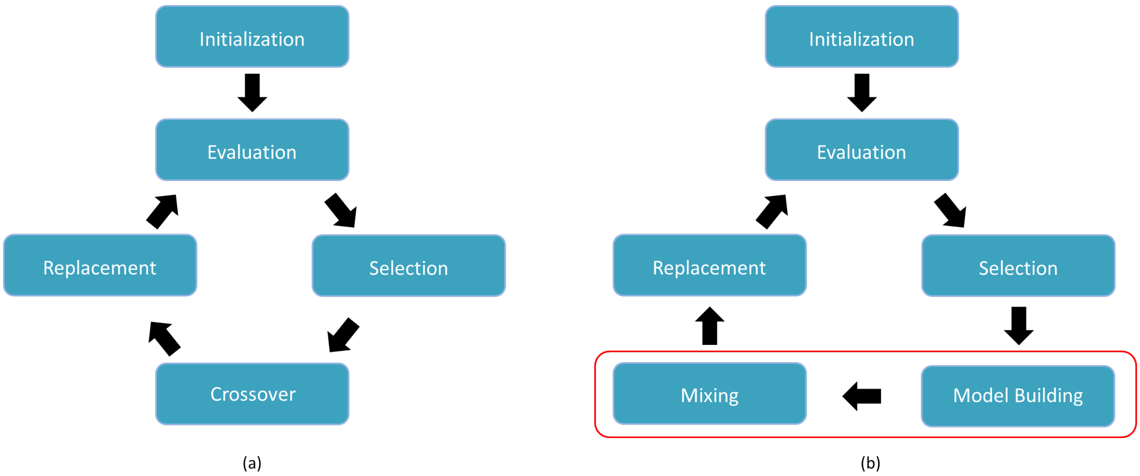


Figure 1.2: The mechanisms of SGA and MBGA.
(a) Simple GA (SGA), which generates offsprings by crossovering random segments. (b) MBGA mixes solutions with substructures identified in the model building.

In 2003, Yu $et al.$ adopted the concept of the dependency structure matrix (DSM) from the organization theory and proposed DSMGA [28]. DSMGA detects and stores the pairwise information in DSM, and the linkage information stored in DSM is used to construct different linkage models. With these models, DSMGA adopts a building-block wise crossover instead of traditional crossover to protect the structures from disruptions. In 2010, Thierens proposed the optimal mixing (OM) operator [21]. In the mixing phase, to generate new solutions, OM donates variables from the parent to the other chromosomes in the population. Unlike the traditional crossover, OM takes the recombination only if the fitness improves. Combining the optimal mixing with the linkage-tree model, the linkage tree genetic algorithm (LTGA) family [1,21,22] has shown strong ability to solve many

doi:10.6342/NTU201704045

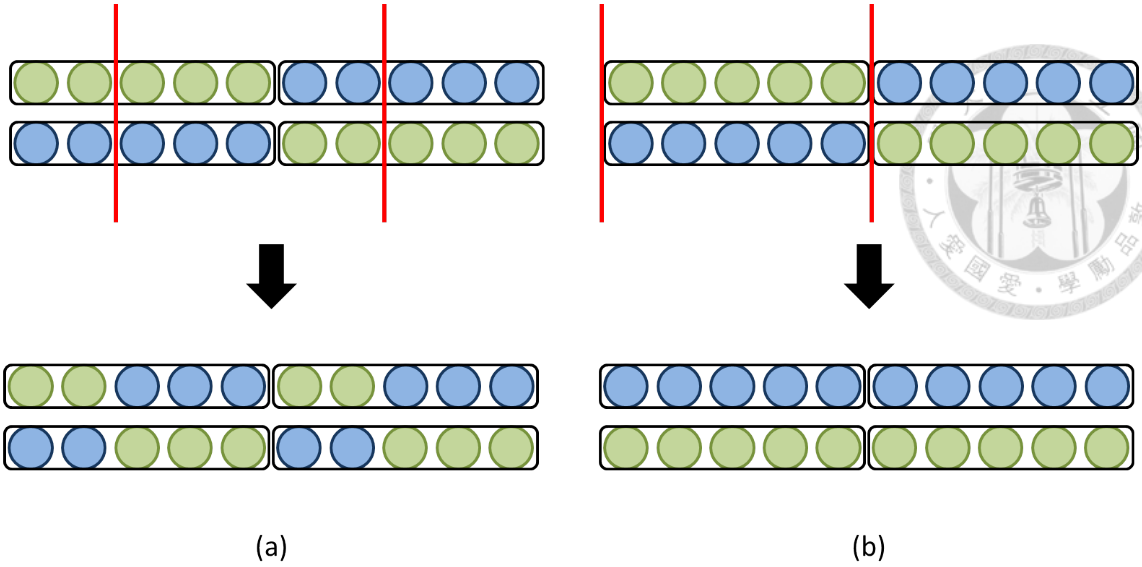


Figure 1.3: The importance of the problem decomposition.
(a) Crossover without structure information. (b) Crossover with structure information.

optimization problems.

Inspired by the optimal mixing, the dependency structure matrix genetic algorithm-II(DSMGA-II) is proposed by Hsu and Yu in 2015 [13]. Based on the DSM used in the DSMGA, 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 the Linkage Tree Genepool Optimal Mixing Evolutionary Algorithm (LT-GOMEA) [1] and hierarchical Bayesian Optimization Algorithm (hBOA) [17], two state-of-the-art MBGAs, on various benchmark problems.

Researches concerning model-building techniques, providing centralized information for recombination, have improved the canonical genetic algorithms by much. However, there is still room for improvement for the model building in MBGAs. In this thesis, I propose a new model which is a customized model for the model building to further improves the performance of DSMGA-II further. The similar concept of customizing has been adopted in the past. For example, consider a simple genetic algorithm with traditional crossover approach to a single machine sequence problem. Crossing over two so-

doi:10.6342/NTU201704045

lutions, in many case, results in an infeasible solution. Many authors developed problem-specific operators such as PMX [8], the order crossover [3], the cycle crossover [16], the subsequence-swap operator [10] and the edge recombination [27] which customize the crossover according to the constrain of the patterns. In recent years, facing with the overlapping problems, Tsuji, based on mincut [29], customize the crossover operator [26] according to the patterns of the parent strings. The operator simplifies the recombinations by removing some edges and nodes according to the alleles of the parent strings before mixing to enhance the efficiency in the crossover. When facing the NK-landscape with overlappings, the partition crossover [25], which contains the similar idea, is adopted in the hierarchical recombinative local search (HiReLS) and the deterministic recombination and iterated local search (DRILS) [2] to avoid exploring many low quality solutions by Francisco in 2017.

Thesis Objectives

In this thesis, based on the idea of the customized techniques described above. I provide the two-edge graphical linkage model which is a customized model for the DSMGA-II as a more effective linkage model. The early stop criterion which slightly enhance the efficiency in mixing is also provided. The modified DSMGA-II we proposed demonstrates stronger optimization ability than the original DSMGA-II on the concatenated trap, folded trap, cyclic trap with 1-bit overlapping, NK-landscape problems with different degrees of overlapping and two real world problems - Ising spin-glass and MAX-SAT.

Roadmap

The rest of the thesis is organized into six parts as follows :

Chapter 2 introduces the concept of DSMGA and OM. The framework of DSMGA-II, and the details of the operators in DSMGA-II are then provided.

Chapter 3 describes the benchmark problems we used in this thesis.

Chapter 4 introduces the concept of the two-edge linkage model in detail, along with a

doi:10.6342/NTU201704045

new issue called supply overfitting which caused by the customized model and some techniques in order to deal with it.

Chapter 5 provides a new termination criterion to enhance mixing effectiveness.

Chapter 6 provides the experiment setup and the results compared with the original DSMGA-II.

Chapter 7 concludes this thesis and describes the directions of future researches involving the work.

doi:10.6342/NTU201704045

國立臺灣大學 · 愛國 · 愛人 · 敬品 · 勵學

doi:10.6342/NTU201704045

Chapter 2

Related Works

For GAs, problems can be solved effectively and efficiently if good patterns of the sub-solutions are combined adequately. In order to handle different kinds of problem structures, many techniques for model building have been developed in the recent years. In this chapter, I first describe two important researches, DSMGA and the optimal mixing, which are incorporated in DSMGA-II. Then I give details of the DSMGA-II which is the main mechanism of our work.

2.1 DSMGA

Borrowed from the concept of the organization theories, the key idea of DSMGA is to extract the building block information by using DSM and clustering algorithms, and then solve the optimization problem more efficiently by using the information of the extracted building block. A DSM is a matrix representation of a graph. The matrix represents the dependency between two variables, which means the pairwise informaion [19]. Each entry e_{ij} is the measure of the dependency between nodes i and j. Entries can be real numbers or integers (Figure 2.1 (a)). The greater the e_{ij} is, the more significant measure between the nodes i and j. After the DSM is constructed, the building blocks information is extracted by clustering the DSM (Figure 2.1 (b)). With the building block information, DSMGA adopts building-block-wise crossover instead of traditional gene-wise crossover in order to reduce the disruptions of the subproblems. Such a mechanism has shown to

doi:10.6342/NTU201704045

be beneficial to the crossover efficiency [28]. The flow of the DSMGA is shown in Figure 2.2.

	A	B	C	D	E
A		0	1	0	0
B	0		0	1	1
C	1	0		0	0
D	0	1	0		1
E	0	1	0	1	

(a)

	A	C	B	D	E
A		1	0	0	0
C	1		0	0	0
B	0	0		1	1
D	0	0	1		1
E	0	0	1	1	

→ {A,C}, {B,D,E}

(b)

Figure 2.1: An illustration of a DSM.
Value 1 represents the linkage exists between two nodes. With certain clustering algorithm, variables can be identified into two disjoint set.

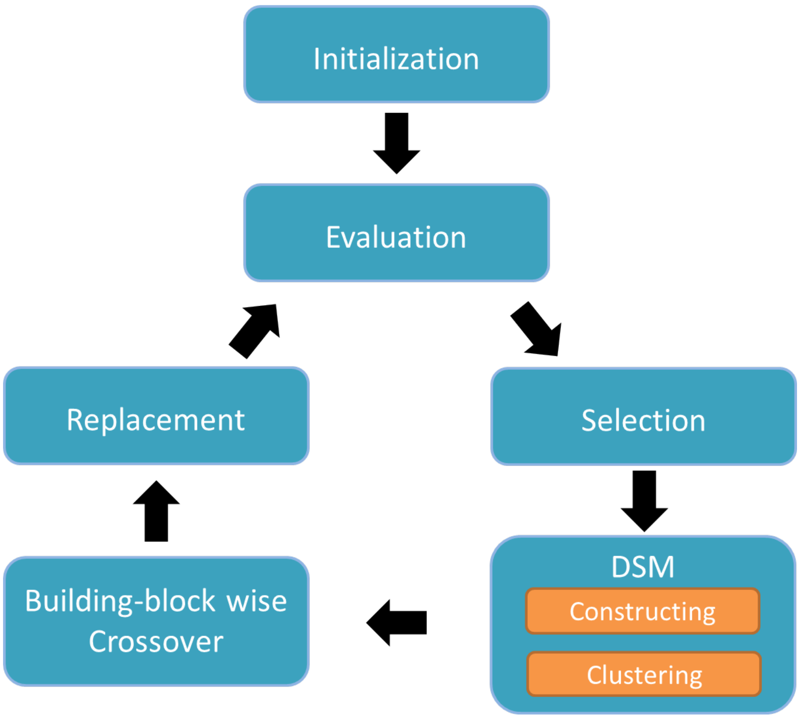


Figure 2.2: The mechanism of DSMGA.

doi:10.6342/NTU201704045

2.2 OM

The optimal mixing evolutionary algorithm (OMEA) was first proposed as the recombinative optimal mixing evolutionary algorithm (ROMEA) and the gene-pool optimal mixing evolutionary algorithm (GOMEA) [22] in 2010. Unlike the canonical GAs, which first proceed the recombination on the parents to generate the offspring and then evaluate the fitness of the offspring, OM applies function evaluations during the recombination and accept the mixing only if the value of the fitness improves (Figure 2.3). The mechanism of the OMEA is shown in Figure 2.4.The structure used in the OM are family of subsets (FOS), denoted as $\mathcal{F}$. Mathematically, a FOS is a set of subsets of a certain main set S. The set S contains all problem variable indexes, $i.e.$, $\{0, 1, ..., \ell - 1\}$. A FOS $\mathcal{F}$ can be written as $\mathcal{F} = \{\mathcal{F}^0, \mathcal{F}^1, ..., \mathcal{F}^{|\mathcal{F}|-1}\}$ where $\mathcal{F}^i \subseteq \{0, 1, ..., \ell - 1\}$ and $i \in \{0, 1, ..., |\mathcal{F}| - 1\}$. Moreover, every problem variable index is contained in at least one subset in $\mathcal{F}$ to ensure no variable is left out of the linkage building process. The pseudo code of GOM variation for FOS model is given in Algorithm 1. OMEAs have shown superior optimization ability to most GAs because of the noise-free decision making in terms of the population sizing and the much smaller required population sizes.

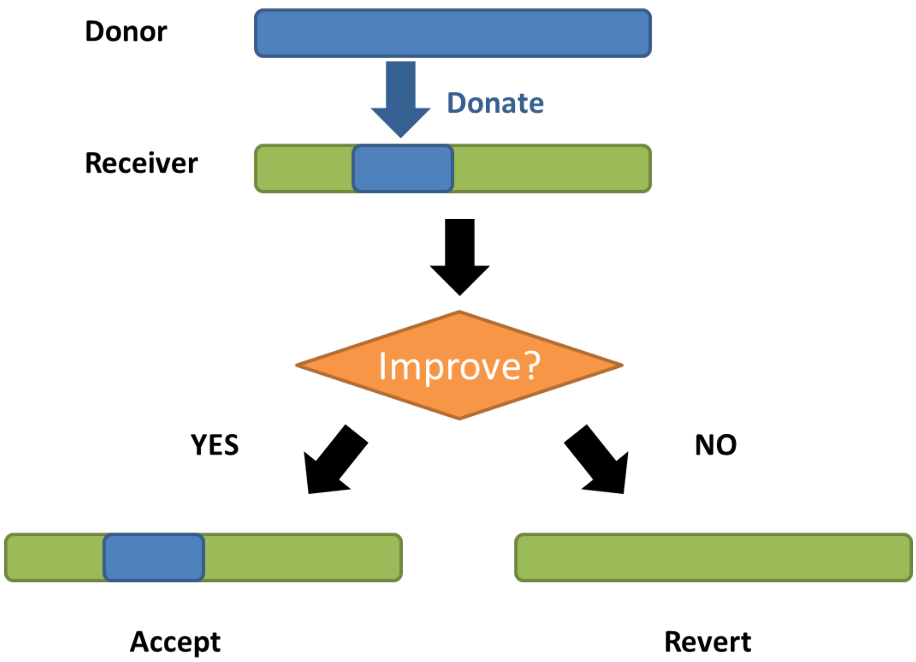


Figure 2.3: The mechanism of OM.
OM accepts the donation only if the recombination leads a improvement.

doi:10.6342/NTU201704045

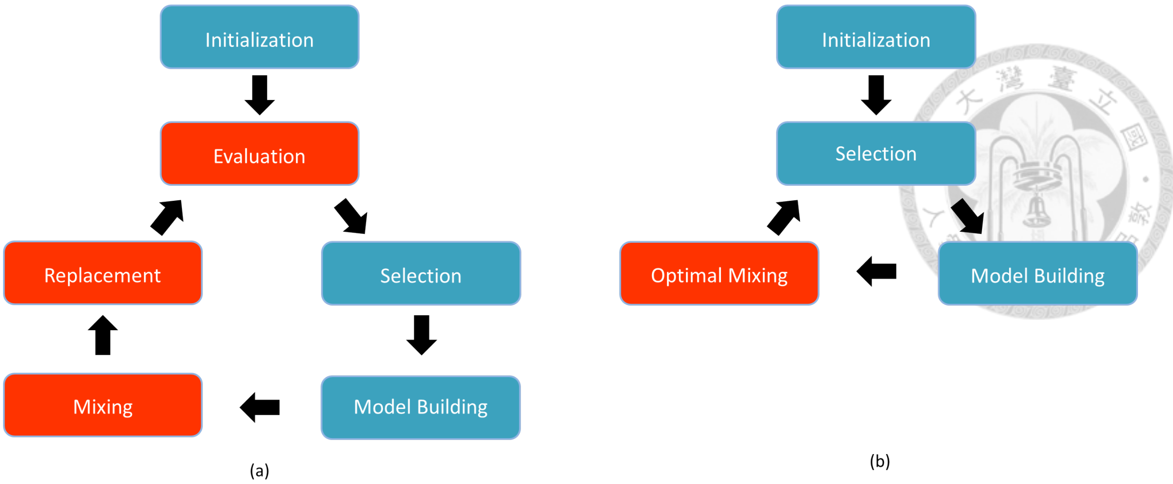


Figure 2.4: The mechanism of OMEA.
OMEA combines three procedures of MBGA (a) into one operator (b).

Algorithm 1: GOM variation for FOS model

P: population, f: evaluation function,
$\mathcal{F}$: FOS, n: population size
input : R: receiver
output: O: offsping

$O \leftarrow R$
$B \leftarrow R$
for $i \leftarrow 0$ **to** $|\mathcal{F}| - 1$ **do**
  $p \leftarrow P_0$ **to** P_{n-1} $O_{F^i} \leftarrow p_{F^i}$ **if** $f(O) \geq f(b)$ **then**
    $B_{F^i} \leftarrow O_{F^i}$
  else
    $O_{F^i} \leftarrow B_{F^i}$
return O

2.3 DSMGA-II

In order to understand the modifications of the dependency structure matrix genetic algorithm II(DSMGA-II) more precisely, the details of the algorithm has to be provided. In this section, I first introduce the framework of DSMGA-II and the concept of incremental linkage set. Then, the pseudo-code of DSMGA-II is provided, and the linkage model is given as well. Finally, I give details of the restricted mixing and the back mixing, which are the kernel operators of DSMGA-II.

doi:10.6342/NTU201704045

2.3.1 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, DSMGA-II performs a bit-flipping greedy hill climbing (GHC) on each chromosome in order to enhance the quality of pairwise linkage model. If the value of the fitness improves, the chromosome accepts the change. Otherwise, the flipped bit would be flipped to original. After population initialization, for the same reason, the chromosomes for the model building are chosen by a tournament selection with selection pressure as 2, suggested in [30].

DSMGA-II adopts mutual information [14] 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. The following steps of the restricted mixing and the 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 the restricted mixing. In the restricted mixing, the receiver tries to flip the bits within the model. If the fitness does not decrease, the receiver becomes the donor in the back mixing and the new pattern is tried on all chromosomes in the population. The population is randomly shuffled before the restricted mixing so that each chromosome takes turns acting as the receiver in the restricted mixing and the donor in the back mixing. The pseudo code of DSMGA-II is given in Algorithm 2 and the flow chart is given in Figure 2.5. The detailed implementations of the ILS and the mixing operators are described in the following sections.

doi:10.6342/NTU201704045

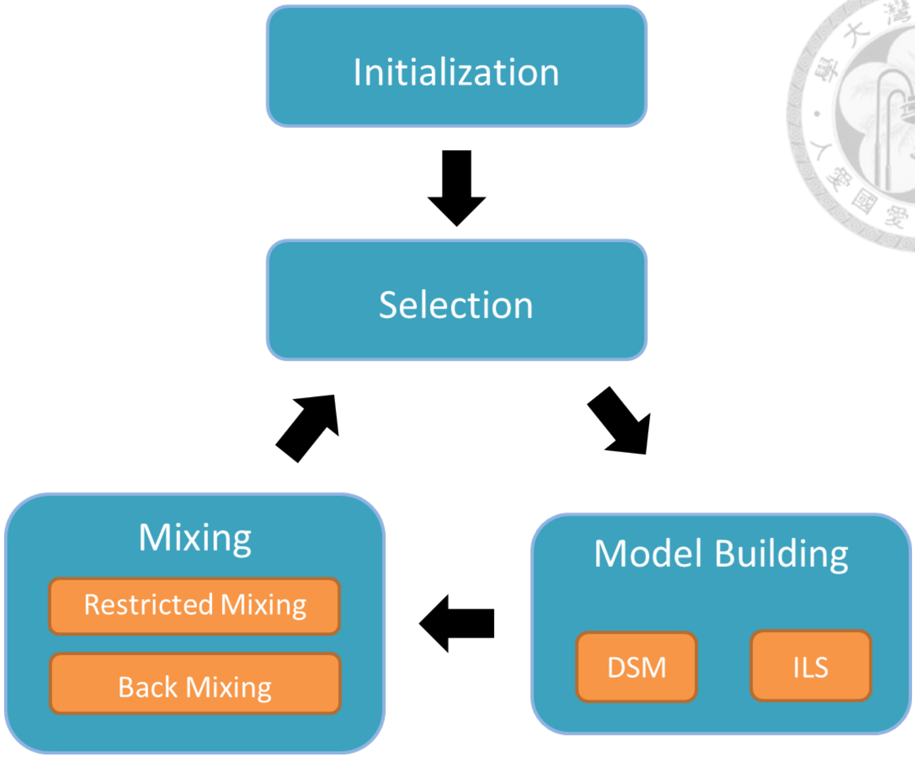


Figure 2.5: The framework of DSMGA-II.

Algorithm 2: 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}

$P \leftarrow \mathrm{PopulationInitialization}(\ell, p)$
$P \leftarrow \mathrm{GHC}(P)$
while *not* ShouldTerminate **do**
  $S \leftarrow \mathrm{TournamentSelection}(P, s)$
  $DSM \leftarrow \mathrm{UpdateMatrix}(S)$
  for $k \leftarrow 1$ **to** R **do**
    $I \leftarrow$ random permutation from 1 to p
    for $i \in I$ **do**
      $P_i, M \leftarrow \mathrm{RestrictedMixing}(P_i)$
      if $M \neq \emptyset$ **then**
        $P \leftarrow \mathrm{BackMixing}(P_i, M)$
return best individual in P

doi:10.6342/NTU201704045

2.3.2 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. Formally, the mutual information of two random variable X and Y can be defined as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}, \tag{2.1}$$

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:

$$I(X;Y) = P_{00} log \frac{P_{00}}{P_{x0}P_{0x}} + P_{11} log \frac{P_{11}}{P_{x1}P_{1x}} + P_{01} log \frac{P_{01}}{P_{x1}P_{0x}} + P_{10} log \frac{P_{10}}{P_{x0}P_{1x}}. \tag{2.2}$$

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 [21] or the average linkage clustering technique proposed in [22], 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 [13]. The equation for finding a node that maximize the weight of AMWCS is defined as:

$$j = argmax_{j \in M'} \frac{1}{|M|} \sum_{k \in M} I(j;k), \tag{2.3}$$

where M is the set of vertices in the current AMWCS, and M' is the set of all vertices but those in M. After insertion, new mask (M) is added into ILS. Therefore, the size of mask becomes larger with insertion time. An example is shown in Figure 2.6.

doi:10.6342/NTU201704045

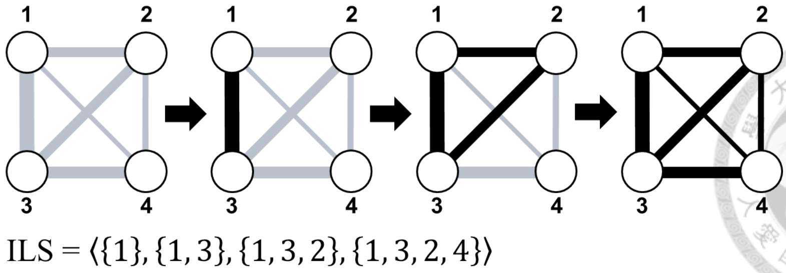


Figure 2.6: An example of the ILS.

2.3.3 Restricted Mixing and Back Mixing

Unlike canonical genetic algorithms that generate offspring by recombining parental solutions, DSMGA-II extends the the idea of OM [22] by adopting two new mixing operators: restricted mixing and back 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 the mixing operations. This way, flipping the bits is equivalent to recombining the receiver with the complement pattern. 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 [9].

After the 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 the recombination, the pattern is accepted and the restricted mixing terminates. Noticed that the operator only accepts the change if the fitness does not decrease. Thus, a noise-free decision making can be achieved with a much smaller population size [7]. The receiver then becomes the donor of the new pattern for the rest of the population in the back mixing. The flow chart is given in Figure 2.7 . The idea behind the restricted mixing is building-block supply [7]. 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 3.

doi:10.6342/NTU201704045

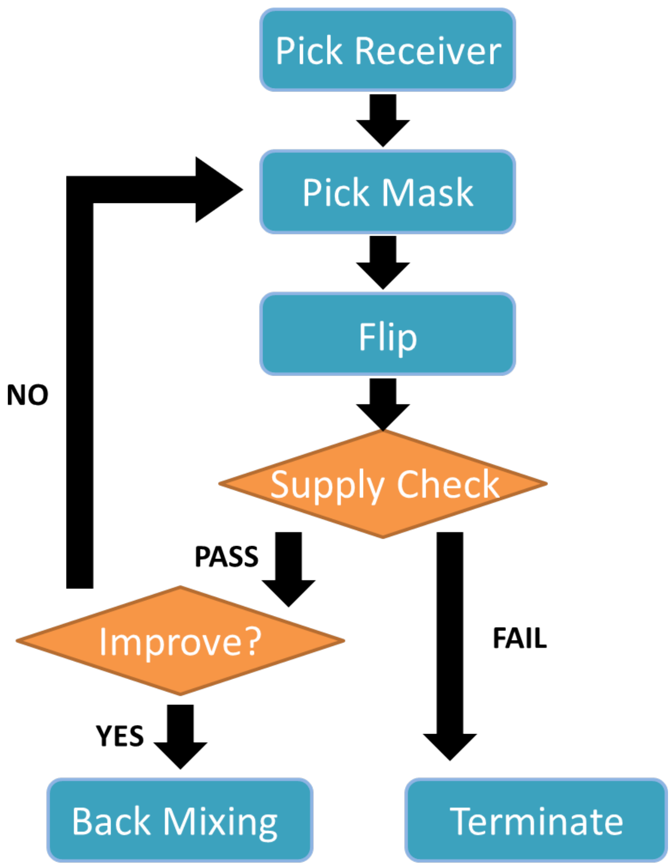


Figure 2.7: The mechanism of the restricted mixing.

doi:10.6342/NTU201704045

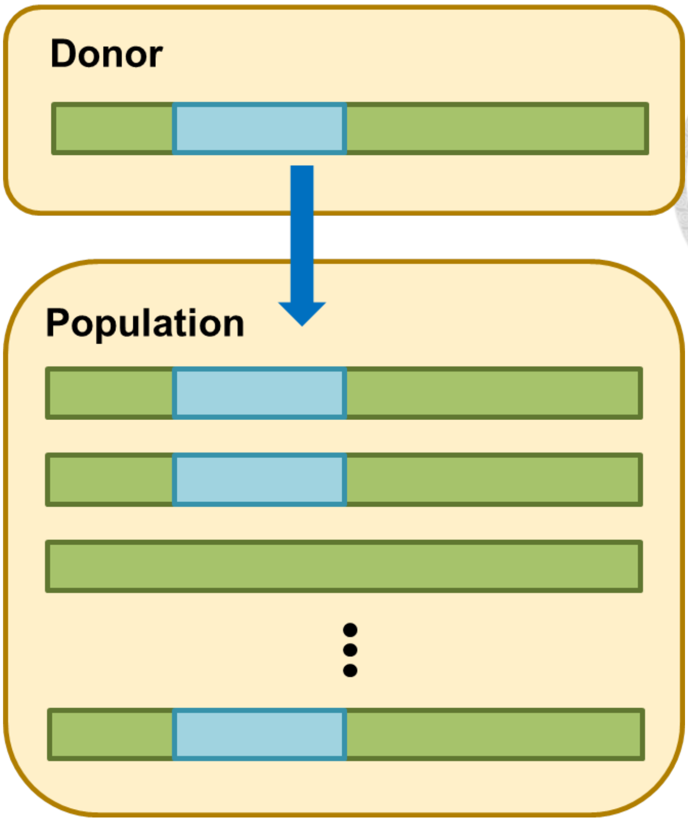


Figure 2.8: The mechanism of the back mixing.

In the back mixing, all the chromosomes in the population are mixed with the pattern accepted in the restricted mixing (Figure 2.8). 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. The back mixing handles the diversity issue with a default strict fitness improvement criterion. When no improvement occurs, it switches to the equal-acceptance criterion 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 of the back mixing is provided in Algorithm 4.

doi:10.6342/NTU201704045

Algorithm 3: Restricted Mixing

P: population,ℓ: problem size,
ILS: incremental linkage set, f: evaluation function,
T: trial solution, M: mask,
R_M: pattern of R extracted by M,
$\overline{R_M}$: complement pattern of R_M
input : R: receiver
output: R: reciever, M: mask

```
for i ← 1 to |ILS| do
    M ← ILS_i
    if R̄_M ⊂ P then
        T ← R
        T_M ← R̄_M
        if f(T) ≥ f(R) and T ∉ P then
            R ← T
            return (R, M)
return (R, ∅)
```

Algorithm 4: Back Mixing

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

```
improved ← false
for j ← 1 to |P| do
    T ← P_j
    T_M ← D_M
    if f(T) ≥ f(P_j) then
        P_j ← T
        improved ← true
    else
        if f(T) = f(P_j) then
            E ← E ∪ {T}
if not improved then
    accept all solutions in E
return P
```


doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Chapter 3

Test Problems

In this chapter, the benchmark problems used in this thesis are introduced. Six types of linkage benchmark problems are considered in my research, including four classic linkage problems (Figure 3.1) and two real-word problems. These benchmark problems each covers different aspects and characteristics of the real-world problems. In following context, the number of variables in the optimization function, which is referred to problem size, is denoted as ℓ , and a chromosome is denoted as a vector $\vec{x} =< x_1, x_2, ..., x_l >$.

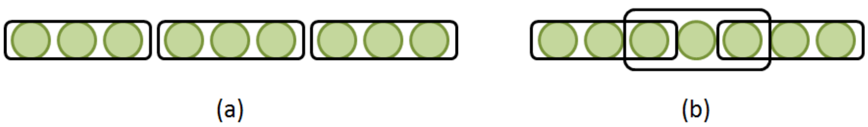


Figure 3.1: Two types of problem structures.
(a)Problem without overlapping structures. (b)Problem with overlapping structures.

doi:10.6342/NTU201704045

3.1 Concatenated Trap

The concatenated trap is composed of m additively separable trap functions. Each trap contains k variables [4]. Figure 3.2 shows the landscape of the (m, k) deceptive trap function. This function is commonly used as benchmark. It is well known that in order to solve trap problems, the underlying structure must be detected and preserved during mixing [23]. If the problem is not properly decomposed, local search is likely to give a solution containing all 0s. The fitness of concatenated trap can be expressed as follows:

$$f_{m,k}^{trap}(\vec{x}) = \sum_{i=1}^{m} f_{k}^{trap}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right), \tag{3.1}$$

where

$$f_{k}^{trap}(u) = \begin{cases} 1 & \text{if } u = k, \\ 0.8 \cdot \frac{k-1-u}{k-1} & \text{otherwise.} \end{cases} \tag{3.2}$$

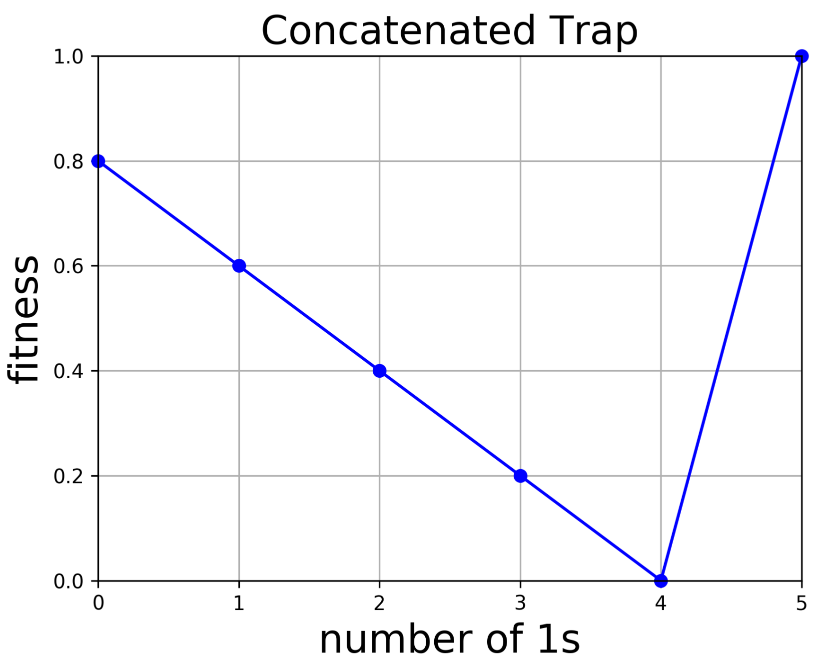


Figure 3.2: Illustration of subproblems of concatenated trap with $k = 5$

.

doi:10.6342/NTU201704045

3.2 Cyclic Trap

The cyclic trap consists of overlapping trap functions with wraparound, and each subproblem is a deceptive trap function. The deceptive trap function is the same as the (m, k) deceptive-trap function in Section 3.1. The fitness of cyclic trap problem can be expressed as follows:

$$f_{m,k}^{cyclic}(\vec{x}) = \sum_{i=1}^{m} f_k^{trap}\left(\sum_{j=i\cdot(k-1)-k+2}^{i\cdot(k-1)+1} x_j\right), \tag{3.3}$$

where

$$f_k^{trap}(u) = \begin{cases} 1 & \text{if } u = k, \\ 0.72 \cdot \frac{k-1-u}{k-1} & \text{otherwise,} \end{cases} \tag{3.4}$$

and

$$x_{\ell+1} = x_1. \tag{3.5}$$

In the cyclic trap, correct problem decomposition is ineffective in some cases. For example, considering a 12-bit cyclic trap with $k = 5$, the fitness of 0000111110000 (one correct subproblem) is lower than 000000000000 (no correct subproblem). This phenomenon increase more trails and NFEs.

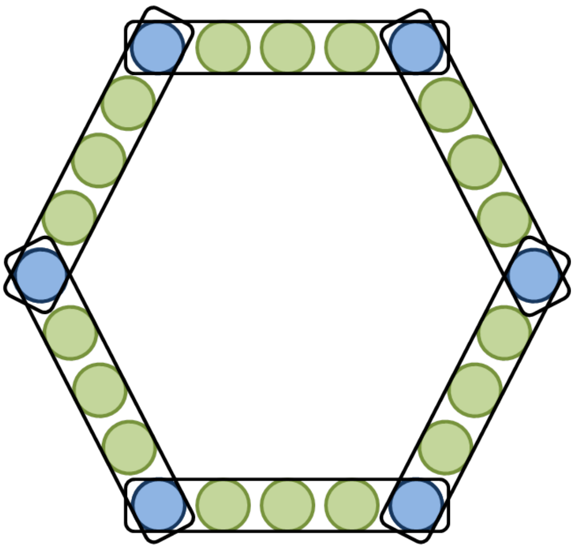


Figure 3.3: Illustration of cyclic trap with $k = 5$ and $l = 24$

.

doi:10.6342/NTU201704045

3.3 Folded Trap

The folded trap is a non-overlapping problem with many local optima on plateaus. In my experiment, I use bipolar deceptive function with $k = 6$. Figure 3.4 shows the landscape of the folded trap function. The folded trap contains two global optima and C_3^6 possible suboptima on the plateaus. Lots of local optima yield unnecessary exploration. Moreover, since the folded trap is symmetric, any preference toward 0s or 1s does not help find the global optimum. The fitness of folded trap problem can be expressed as follows:

$$f_{m,k=6}^{folded}(\vec{x}) = \sum_{i=1}^{m} f_{k=6}^{folded}\left(\sum_{j=i\cdot k-k+1}^{i\cdot k} x_j\right), \tag{3.6}$$

where

$$f_{k=6}^{folded}(u) = \begin{cases} 1 & \text{if } |u-3| = 3, \\ 0 & \text{if } |u-3| = 2, \\ 0.4 & \text{if } |u-3| = 1, \\ 0.8 & \text{if } |u-3| = 0. \end{cases} \tag{3.7}$$

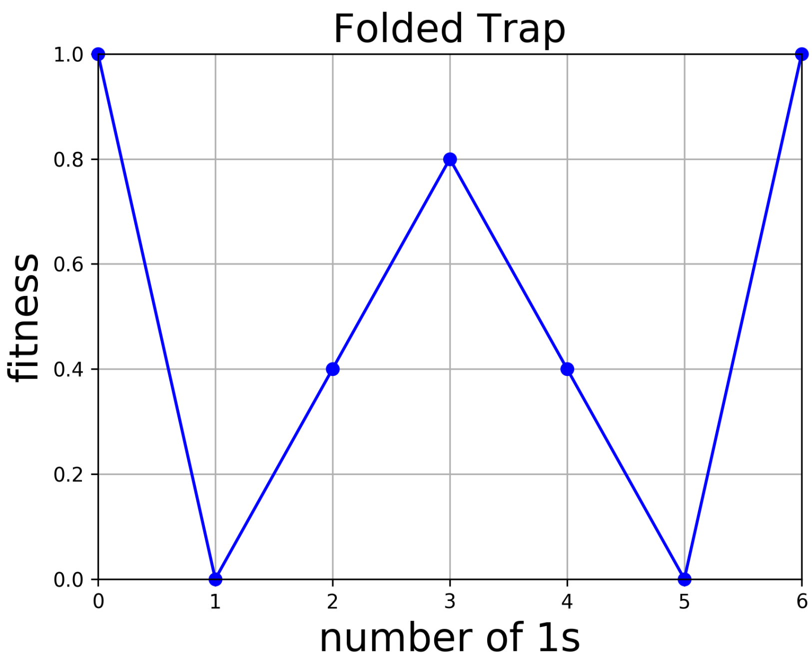


Figure 3.4: Illustration of subproblems of folded trap with $k = 6$

.

doi:10.6342/NTU201704045

3.4 NK-landscape

The NK-landscape problem consists of overlapping, randomly generated subfunctions. There are three parameters of the function: 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. In my experiments, I use $k = 4$ and NK-landscape with steps 1, 3 and 5 to represent problems with different degrees of overlapping. The fitness of NK-landscape problem can be expressed as follows:

$$f_{m,k}^{NK}(\vec{x}) = \sum_{i=1}^{m} f_{i,k}^{NK}(x_{i\cdot k-k+1}, x_{i\cdot k-k+2}, ..., x_{i\cdot k-k+k}), \tag{3.8}$$

Because of the randomly generated landscape, the NK-landscape functions are considered as general cases of problems.

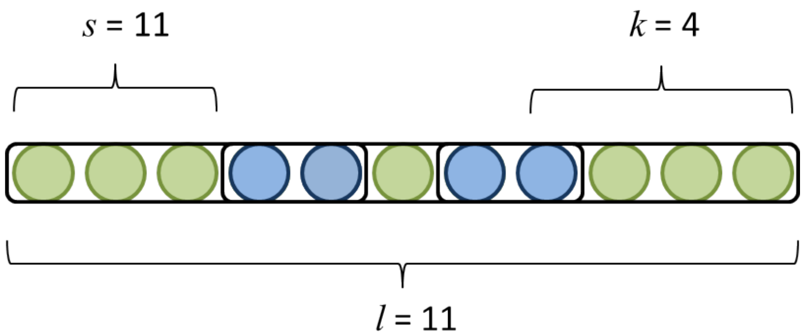


Figure 3.5: Illustration of NK-S3 with $k = 4$ and $\ell = 11$

.

doi:10.6342/NTU201704045

3.5 Ising Spin-glass

The Ising model, named after the physicist Ernst Ising, is a mathematical model of ferromagnetism in statistical mechanics. The Ising spin-glass gives a set of variables in one of the two states $\{+1, -1\}$. For each pair of adjacent 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 fitness of Ising Spin-glass can be expressed as follows:

$$f_n^{spin}(x) = -\sum_{i,j=0}^{n} x_i x_j J_{ij}. \tag{3.9}$$

Ising spin-glass systems are usually studied for their particular property, such as plateaus and symmetry. In my experiment, I use 2-D Ising spin-glass which is one of the simplest statistical models to show a phase transition.

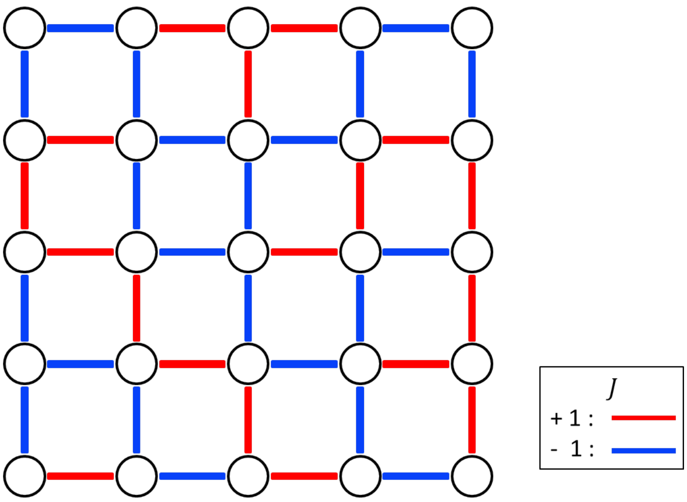


Figure 3.6: Illustration of 2-D Ising spin-glass with 5x5 grid structure.
Notice that the structures are all wraparound.

doi:10.6342/NTU201704045

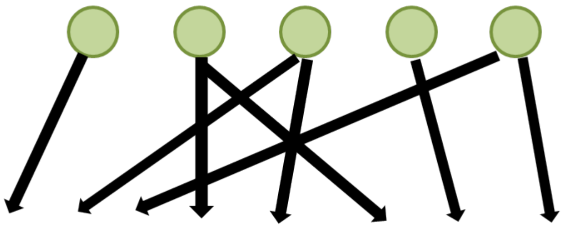
3.6 MAX-SAT

In computational complexity theory, the maximum satisfiability problem (MAX-SAT) is a classical NP-complete problem [24]. It consists of a series of logical and clauses. Each clause contains a series of logical or variables. The MAX-SAT can be describe in conjunctive normal form as follows:

$$F = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{k_i} \ell_{ij} \right), \tag{3.10}$$

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.

The MAX-SAT is a generalization of the Boolean satisfiability problem, which is the problem of determining whether the variables of a given Boolean formula can be assigned in such a way as to make the formula evaluate to TRUE. In my experiment, I use the Uniform Random-3-SAT instances from SATLIB[1] with all satisfiable clauses.



$$(\neg X_1 \lor X_3 \lor X_5) \land (X_2 \lor X_3) \land (\neg X_2 \lor \neg X_4 \lor X_5)$$

Figure 3.7: Illustration of MAX-SAT with 3 clauses and 4 variables.

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

doi:10.6342/NTU201704045

學大灣臺立國 人愛國愛・學勵品敦

doi:10.6342/NTU201704045

Chapter 4

Two-edge Graphical Linkage Model

In the original AMWCS, two nodes are connected with only one linkage. I call this technique the one-edge graphical linkage model. In this chapter, I first describe the new linkage model, called the two-edge graphical linkage model, which gives the customized recombination mask for each chromosome. Then I discuss the problem of supply overfitting caused by the new model. Finally, I gives two attempts to deal with the overfitting problem in details.

4.1 Two-edge Graph

In the scheme of the 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:

$$L(00 \cup 11) = P_{00} log \frac{P_{00}}{P_{0x}P_{x0}} + P_{11} log \frac{P_{11}}{P_{1x}P_{x1}}, \tag{4.1}$$

and

$$L(01 \cup 01) = P_{01} log \frac{P_{01}}{P_{0x}P_{x1}} + P_{10} log \frac{P_{10}}{P_{1x}P_{x0}.} \tag{4.2}$$

The reason for such division is to protect the pattern within each building block. For the 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

doi:10.6342/NTU201704045

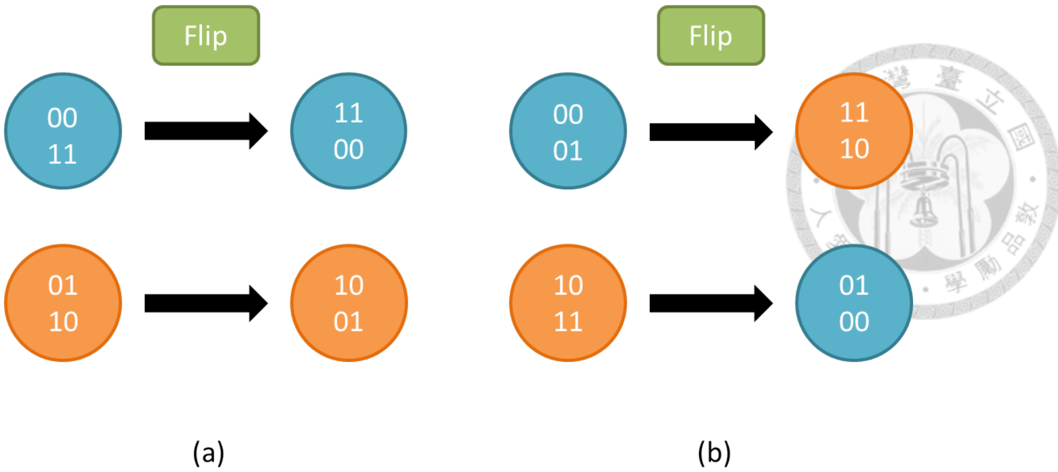


Figure 4.1: Illustration of the reason for such division.
(a) and (b) show the reason for dividing the patterns into $\{00, 11\}$ and $\{01, 10\}$. After flipping, the division in (a) is same as before. But the division in (b) is staggered after flipping.

to different chromosomes after mixing, the original patterns are reserved in population. Same scenario happens with pattern 01 and 10. Figure. 4.1 shows the problem based on a different way of division.

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

$$j = argmax_{j \in M'} \frac{1}{|M|} \sum_{k \in M} I(j;k), \tag{4.3}$$

where M is the set of vertices in the current AMWCS, and M' is the set of all vertices but those in M.

This equation is 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 the model construction. The rule of the edge selection in the two-edge graph is as follows:

$$L(X;Y) = \begin{cases} L(00 \cup 11) & \text{if pattern is 00 or 11,} \\ L(01 \cup 10) & \text{if pattern is 01 or 10.} \end{cases} \tag{4.4}$$

Figure 4.2 depicts different results of model construction with a problem of three bits. The node 1 is randomly chosen from the candidate set $\{1, 2, 3\}$ in the beginning. For the

doi:10.6342/NTU201704045

one-edge scheme in Figure 4.2(a), node 3 is first selected according to the strongest edge $\{1, 3\}$. After two iterations, the inserted node sequence of the AMWCS is $Q = \langle\{1, 3, 2\}\rangle$, and the ILS is $\langle\{1\}, \{1, 3\}, \{1, 3, 2\}\rangle$.

For the two-edge scheme, we first consider a receiver with alleles $\{0, 0, 1\}$ in Figure 4.2(b), although there is a strong blue edge between nodes 1 and 3, the pattern 10 conflicts with the meaning of the light blue edges $L(00 \cup 11)$. According to the linkage selection rule, the thin red edge $L(01 \cup 10)$ represents the linkage between nodes 1 and 3 with pattern 01. The graph after removing all the conflict edges for clarity is represented in Figure 4.2(c). Unlike the one-edge scheme, node 2, with the strongest edge $\{1, 2\}$, is picked in Figure 4.2(c). The inserted node sequence of the AMWCS is $Q = \{1, 2, 3\}$, and the ILS is $\langle\{1\}, \{1, 2\}, \{1, 2, 3\}\rangle$. In Figure 4.2(d), a receiver with alleles $\{0, 1, 0\}$ and a same 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 $\langle\{1\}, \{1, 3\}, \{1, 3, 2\}\rangle$ is the same as one-edge graph in Figure 4.2(a), the two-edge graph is only constructed with non-conflict edges.

Consider an optimal subsolution 111 which structure is similar to the three-bit problem in Figure 4.2. Suppose that strong linkages can be detected among these three bits, following the procedures described above, the one-edge graph constructs the ILS as $\langle\{1\}, \{1, 3\}, \{1, 3, 2\}\rangle$. However, none of mask in the ILS can flip the receiver with pattern 001 to 111. In short, even if the one-edge scheme detects the correct model, it might still fail during mixing. On the other hand, two-edge graphical linkage model handles this problem by taking the alleles of receivers into account and preserving the correct pattern during the restricted mixing. For the two-edge scheme, choosing the mask $\{1, 2\}$ (Figure 4.2(c)) can help flipping the pattern 001 to the the optimal subsolution 111. Moreover, if the pattern of the receiver is 010, the mask $\{1, 3\}$ (Figure 4.2(d)) can still flip the pattern 010 to 111.

The two-edge model tends to align the alleles of the receiver with the dominant patterns in the population because of one of the characteristic of the mutual information. Since the mutual information between two bits is divided into two parts in the two-edge scheme, the linkage which containing the high-ratio pattern is always positive, while the other part is

doi:10.6342/NTU201704045

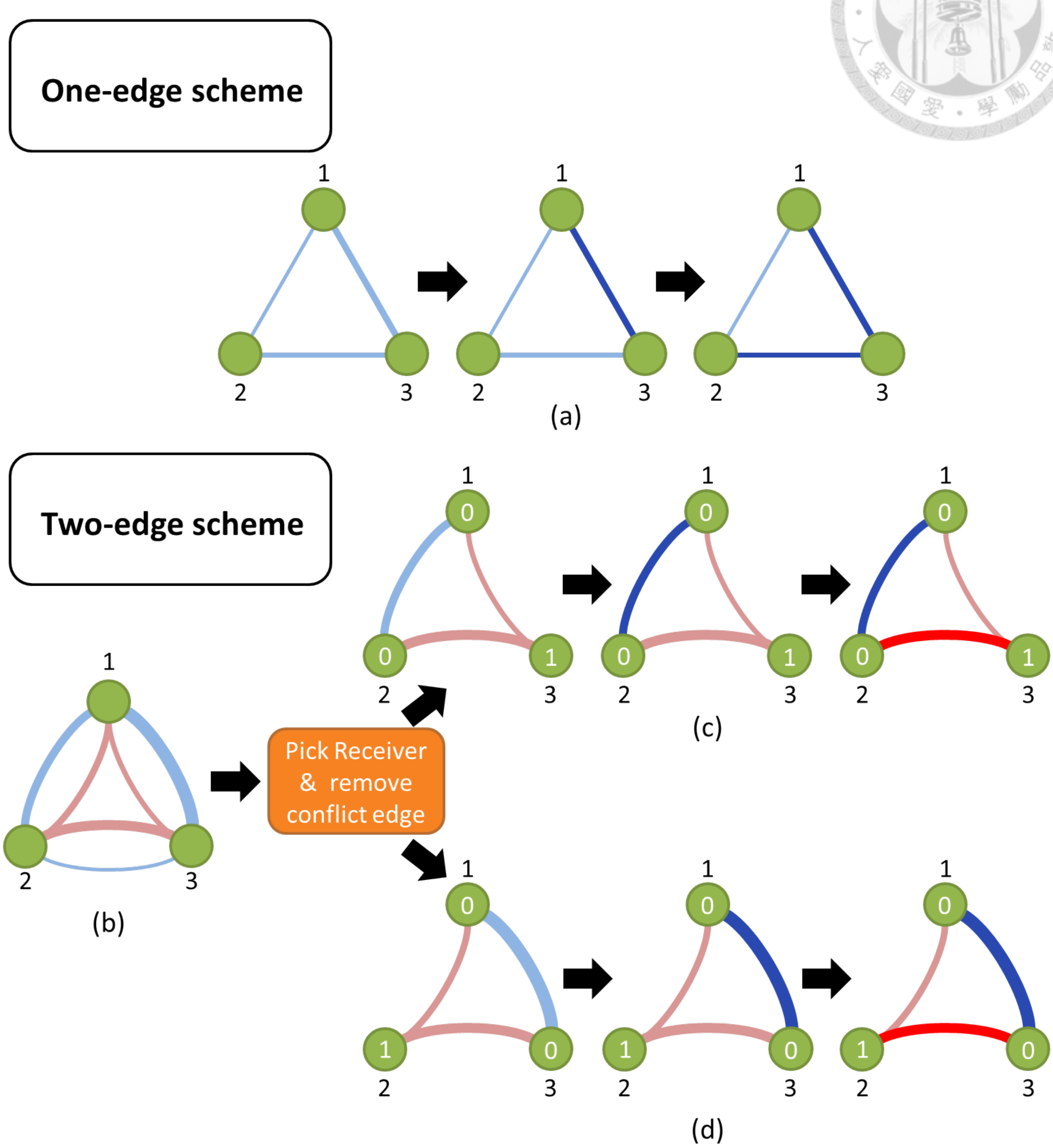


Figure 4.2: Illustration of the construction of one-edge model and two-edge model. The width of the edge corresponds to the strength of the linkage. The value in each node is the allele of the receiver. Notice that the alleles are not considered in the one-edge scheme. For (b) to (d), the light blue and blue edges represent the L(00∪11) edge, while the pink and red edges represent the L(00∪11) edge. Nodes and edges with color blue or red represent the determined AMWCS. (c) and (d) represent different construction according to the different receiver.

doi:10.6342/NTU201704045

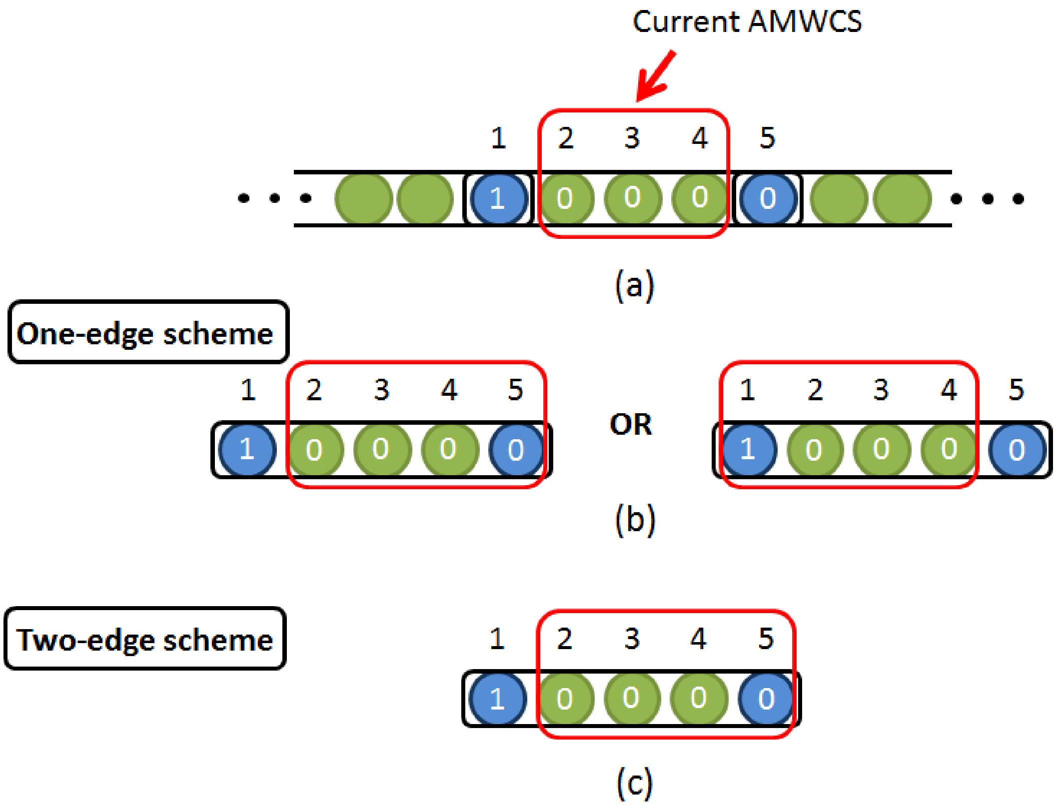


Figure 4.3: Illustration of the different sequence of AMWSC between the one-edge and the two-edge model.

(a) shows that after 2 iterations, the AMWCS is $\{2, 3, 4\}$. (b) and (c) represent the problem of inaccurate insertion for the one-edge scheme.

often negative. For the example in Figure 4.2, 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 such case, the two-edge model always provides the correct recombination mask for the different receivers.

For another example, consider a subproblem of the cyclic trap with $k = 5$ and AMWCS constructed as $\{2, 3, 4\}$ in Figure 4.3. Notice that after GHC, the ratio of patterns 00001 and 10000 are the same in average. For the one-edge scheme, both nodes have a fifty-fifty chance depended on the current population of being chosen next. If the node 1 is chosen, the restricted mixing fails. For the two-edge model, because the ratio of the pattern 00000 dominates, the next node chosen by the two-edge scheme is likely to be 5 but not 1 according to the property described above. With the mask $\{2, 3, 4, 5\}$, the scheme can flip the pattern 10000 to pattern 11111.

doi:10.6342/NTU201704045

4.2 Supply Overfitting

The customized model seems promising. However, I notice that the two-edge scheme costs even more evaluations in the mixing stage. The results are shown in Table 4.1. Although the result is frustrating, instead of giving up the idea of the customized model, I modify the model building procedure and conduct an experiment. I adopts the one-edge model in the beginning. Then I attempt to switch the one-edge model to the two-edge model at different timings. The result (Figure) shows that if we adopts the two-edge model in the late stage, the two-edge model leads a 1% to 4% improvement. Although the improvement is pretty small, it is significant for me to show that the idea of the customized model works. After further investigation into the bad performance in the early stage, I find that 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, I choose the node that gives the maximum linkage during the 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 patterns that are customized for a specific receiver has a greater chance to pass the supply check. The relationship between the supply length and the linkage strength becomes a "chicken-egg" problem.

For example, Figure 4.4 demonstrates the different supply length between the one-edge model and the two-edge model for a receiver with the pattern 101111. The supply length of one-edge model is only two since the complement pattern 01 does not exist in the population. On the other hand, since the two-edge model is customized for the receiver, it has a higher possibility in finding the complement patterns. In this case, the supply length of two-edge model is six which is much more than the supply length of the one-edge model. Compared to the model constructed by the one-edge graph, the masks in the two-edge ILS are more likely to pass the supply check.

The experiment results in Figure 4.5 show that the average supply length of the two-edge model is about twice longer than the one-edge counterpart in most of the problems.

doi:10.6342/NTU201704045

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

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

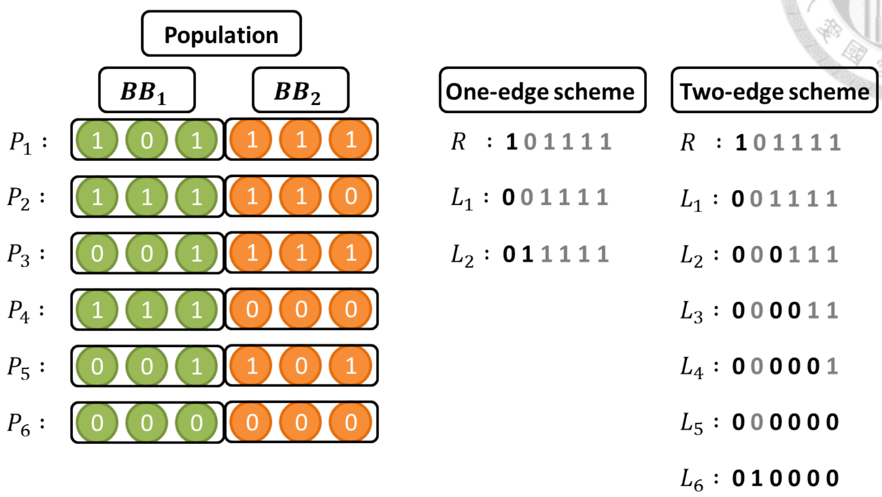


Figure 4.4: An example for the overfitting supply length

.

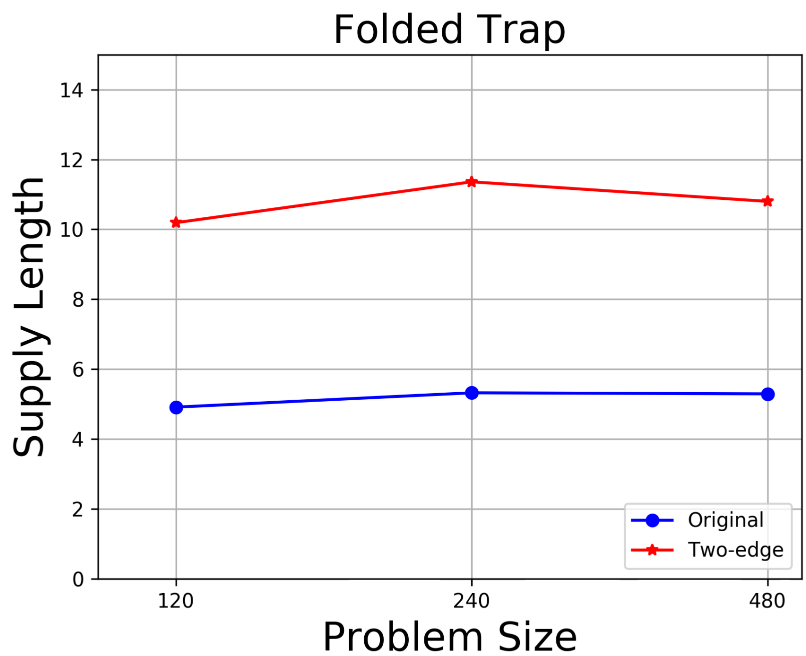


Figure 4.5: The different supply length in average between different models.

doi:10.6342/NTU201704045

Generally speaking, the supply length prohibits recombination of longer patterns. However, in the back mixing, when using the supply length which is overfitting, it is likely to try larger patterns and copy a larger portion of the donor to all chromosomes in the population. Since the information in the early stage is not enough, these attempts usually increase failure trails and the probability of cross-competition in back the mixing.

On the other hand, although the customized model has some advantages in the restricted mixing, the overfitting supply length has a negative impact on the restricted mixing. For example, in Figure 4.4, both models, in the end, fail in the restricted mixing, but the one-edge scheme terminates the restricted mixing earlier than the two-edge scheme because of the shorter supply length, which means the one-edge scheme reduces more NFEs in the restricted mixing. Thus, the overfitting supply length causes much more unnecessary trails in the restricted mixing as well. To sum up, in the early stage in DSMGA-II, since the information and the signals in the population are not clear enough, the linkage detections are often inaccurate. Under this circumstance, choosing shorter pattern for the mixing seems to be a better choice because of the high failing rate in the mixing stage.

4.3 Representative Supply Check

To avoid supply length from overfitting the donor, I provide an idea called representative supply check, which is a modified version of the original supply check in the DSMGA-II. For the original version, even though only one complementary pattern of the receiver exists in the population, the mask still passes the supply check. This situation exists frequently

Problems	Original	Modified	Ratio
Concat. trap, $\ell = 400$	55.0	166.3	-202.3%
Cyclic trap, $\ell = 400$	123.1	114.8	6.8%
Folded trap, $\ell = 480$	354.5	334.6	5.6%
NK-S1, $\ell = 400$	767.4	1021.2	-33.0%
NK-S3, $\ell = 400$	550.5	1004.7	-82.4%
NK-S5, $\ell = 400$	62.1	76.5	-23.1%
2D spin-glass, $\ell = 400$	245.2	288.2	-17.5%
MAX-SAT, $\ell = 100$	221.6	1363.5	-515.3%

Table 4.1: Required NFE of DSMGA-II for the largest test problems (unit: K)

doi:10.6342/NTU201704045

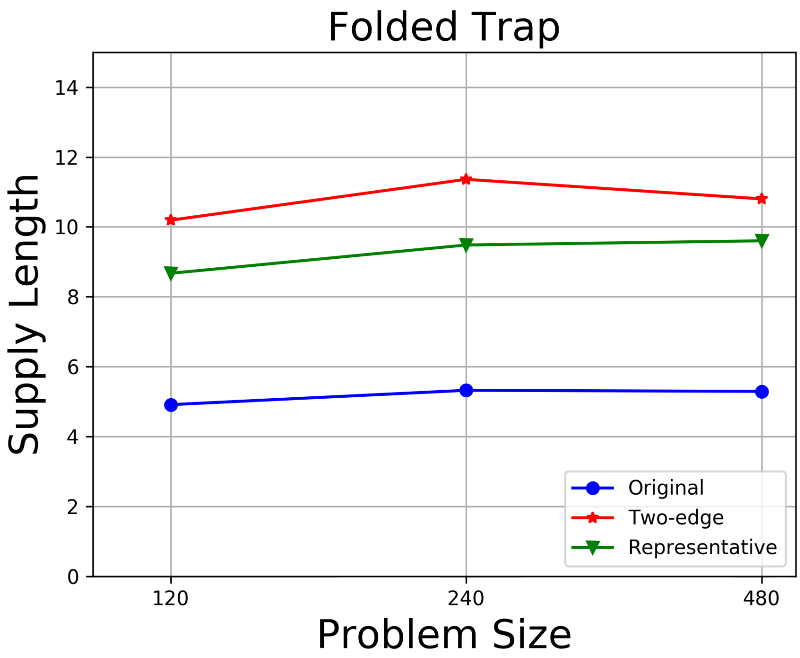


Figure 4.6: The different supply length in average between different models.

in the two-edge model due to the problem of supply overfitting. Since I believe that the ratio of the prevailing pattern increase gradually during the mixing stage, if the population contains just only one complementary pattern, the pattern for recombination is unlikely to be taken during the mixing. In contrast, if the population contains a high ratio of a pattern, this pattern is likely to be accepted by the population during the back mixing. Thus, the concept of the representative supply check is that the mask passes the check only if the population contains a specific numbers of the complementary pattern.

However, the modified supply check just slightly shorten the supply length. The result is shown in Figure 4.6. Even worse, I find that this scheme gives disastrous results in some cases. For example, considering the cyclic trap problem shown in Figure 4.7(a), with the original supply check, although the pattern 11111 may be flipped to 00000 if I choose this chromosome as the receiver, the probability is only 6% or so. In contrast, it is more likely to pick the chromosome with the pattern 00000 and flip the pattern 00000 to the optimal pattern 11111. But considering the representative supply check in Figure 4.7(b). Since the ratio of the pattern 11111 is usually low, the pattern 11111 is hard to pass the representative supply check. Thus, even I choose the chromosome with the pattern 00000 as the receiver, the pattern 00000 is hard to be flipped to the pattern 11111 owing to the supply check fail.

doi:10.6342/NTU201704045

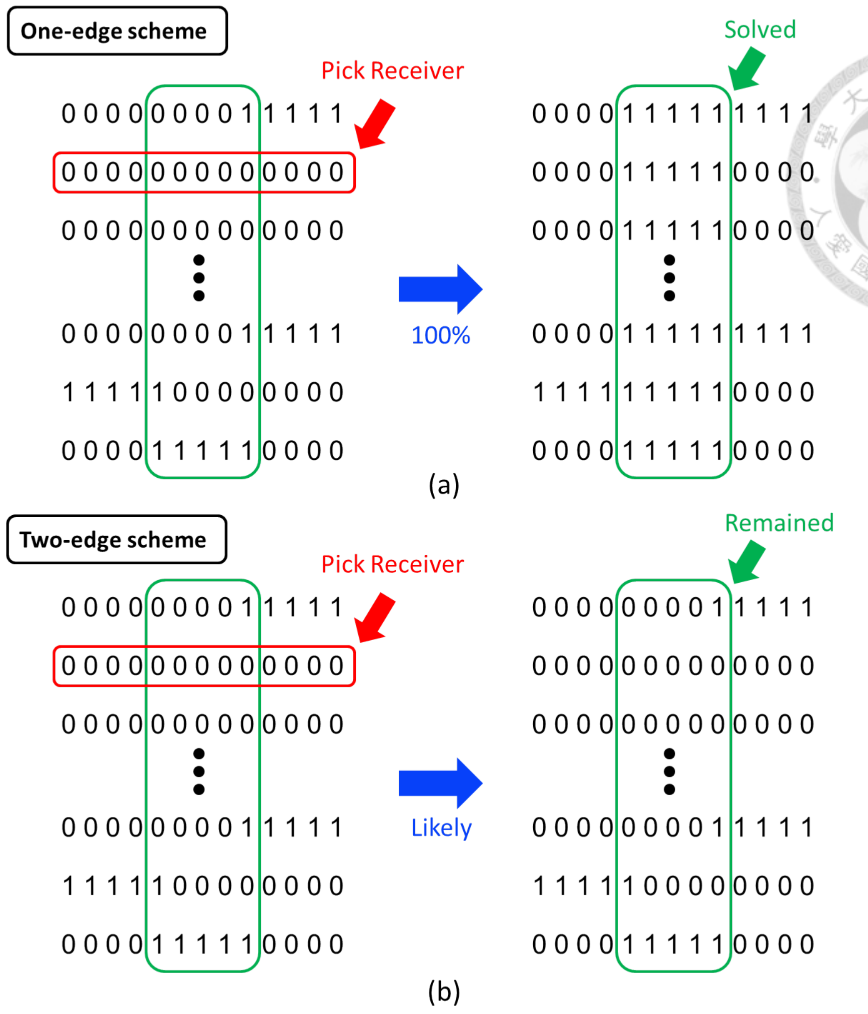


Figure 4.7: Mixing for the cyclic trap with the original and the modified supply check.

In other words, adopting the representative supply check, in such case, is very likely to being stuck in the local optimum. Generally speaking, this scheme potentially increases the competitions between the local and the global optimum in some cases.

4.4 One-edge Supply Bound

Considering the supply lengths provided by the different models, the shorter supply length which provided by the one-edge scheme seems to be more beneficial to the mixing as described in the Section 4.2. Besides, for example, considering the folded trap problem, we know that the mixing works more effectively and efficiently by adopting the supply length which is less than 6 (e.g., 2, 3, 4). The average supply length which satisfy this ground truth (Figure 4.6) is only provided by the one-edge model. Since the one-edge

doi:10.6342/NTU201704045

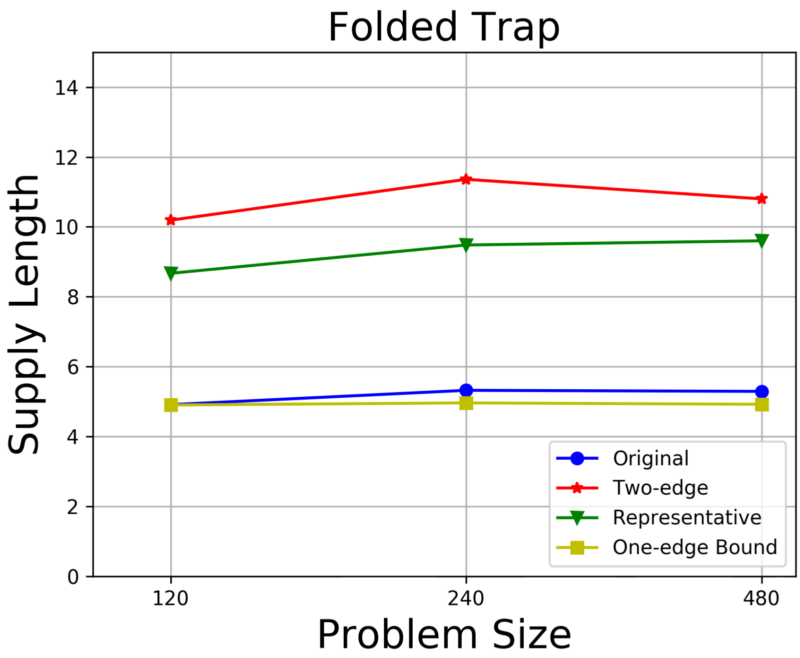


Figure 4.8: The different supply length in average between different models.

model provides the supply length which seems to be a better choice as described above, I attempt to combine the supply length given by the one-edge model with the two-edge graph. This idea is called *supply bound*. During the restricted mixing with applying this idea, after picking a receiver, I first construct the one-edge linkage model, then I build the ILS according to the one-edge graph for the recombination masks. All masks in the ILS must go through the supply check. Finally, the supply bound is produced according to the longest mask which passes the original supply check. With the supply bound, if the recombination mask constructed by the two-edge scheme is longer than the supply bound, the mask is cancelled out. The flow of the restricted mixing with one-edge bound is shown in Figure 4.9, and the pseudo code is given in Algorithm 5.

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 measured in the two-edge graph is more skewed which favors the given receiver during the 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. The supply length compared with other schemes described above are shown in Figure 4.8.

doi:10.6342/NTU201704045

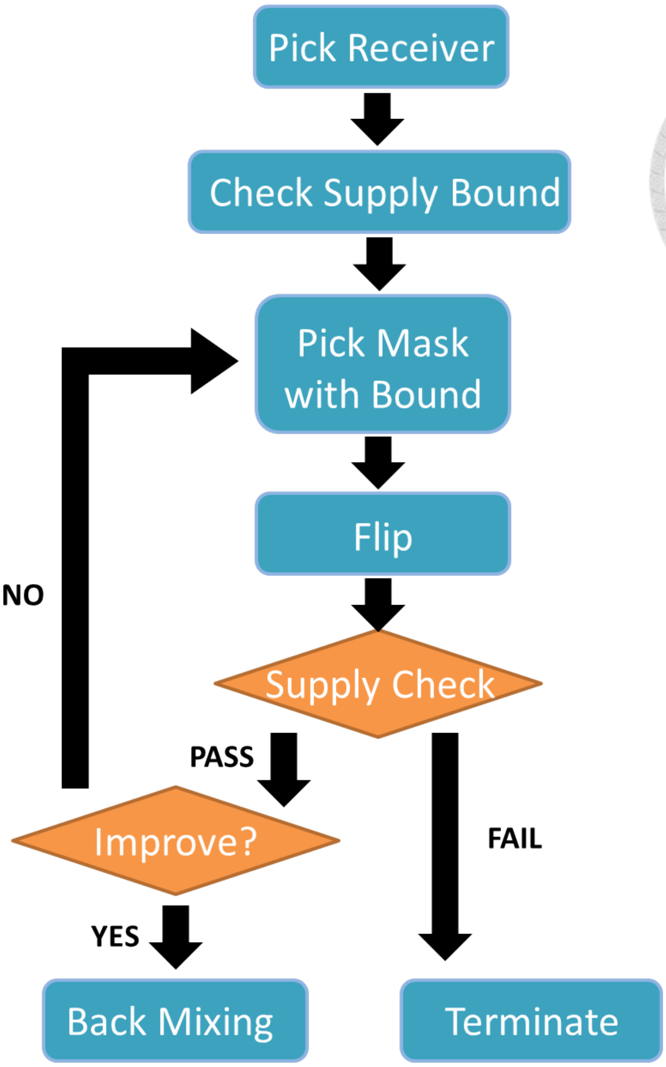


Figure 4.9: The mechanism of the restricted mixing with one-edge bound.

As mentioned before, the supply length provided by the one-edge model seems to benefit the mixing operators. Especially in the early generations, subproblem patterns with better fitness do not stand out. The dependency information in the population is unclear, which results in building the inaccurate models. The inaccurate models then increase the failures in the mixing stage. With the masks bounded by the one-edge graph supply length, unnecessary recombination trials can be reduced during the mixing stage. The result of required NFE of DSMGA-II which combines the two-edge model with the one-edge supply bound is shown in Table 4.2.

doi:10.6342/NTU201704045

Algorithm 5: Modified Restricted Mixing with One-edge Bound

ILS_one: one-edge incremental linkage set,
ILS_two: two-edge incremental linkage set,
f: evaluation function,
P: population, ℓ: problem size,
T: trial solution, M: mask,
R_M: pattern of R extracted by M,
$\overline{R_M}$: complement pattern of R_M
input : R: receiver
output: R: reciever, M: mask
for $i \leftarrow 1$ **to** $|ILS\_one|$ **do**
  $M \leftarrow ILS\_one_i$
  if $\overline{R_M} \not\subset P$ **then**
    $SupplyBound \leftarrow i$

for $i \leftarrow 1$ **to** $|ILS\_one|$ **do**
  $M \leftarrow ILS\_two_i$
  if $\overline{R_M} \subset P$ **then**
    $T \leftarrow R$
    $T_M \leftarrow \overline{R_M}$
    if $f(T) \geq f(R)$ **then**
      $R \leftarrow T$
      return (R, M)

return (R, $\emptyset$)

doi:10.6342/NTU201704045

Problems	Original	Modified	Ratio
Concat. trap, $\ell = 400$	55.0	54.7	0.5%
Cyclic trap, $\ell = 400$	123.1	116.4	5.4%
Folded trap, $\ell = 480$	354.5	240.4	32.2%
NK-S1, $\ell = 400$	767.4	723.0	5.7%
NK-S3, $\ell = 400$	550.5	548.2	0.4%
NK-S5, $\ell = 400$	62.1	59.8	3.6%
2D spin-glass, $\ell = 784$	863.2	709.9	17.8%
MAX-SAT, $\ell = 200$	7887.6	6346.0	19.5%

Table 4.2: Required NFE of DSMGA-II for the largest test problems (unit: K)

doi:10.6342/NTU201704045

Chapter 5

Early Stop Criterion

In this chapter, I propose a new stop criterion for the restricted mixing which eliminates unnecessary trials and slightly 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 a bit-flipping with the next mask. This bit-flipping process only terminates 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.

For example, consider a five-bit trap problem with global optimum 11111 and local optimum 00000 in Figure 5.1. P_1 is chosen as receiver in restricted mixing. Suppose the

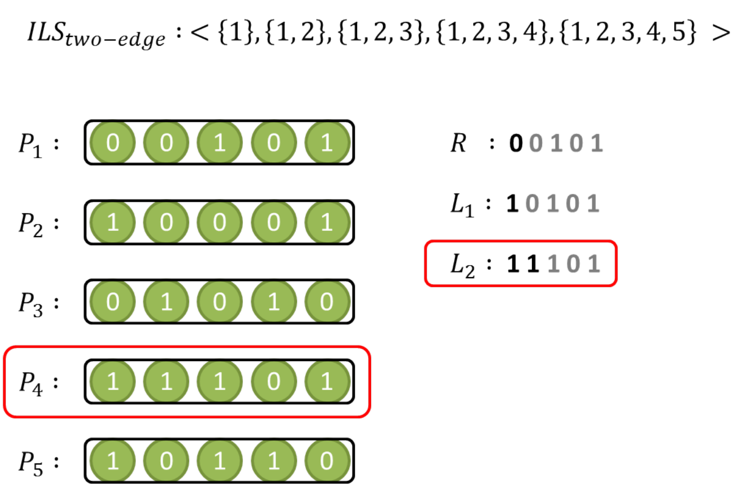


Figure 5.1: The mechanism of the restricted mixing with one-edge bound.

doi:10.6342/NTU201704045

masks in ILS follow the sequence $\{1, 2, 3, 4, 5\}$, and the complement patterns for all five models exist in the population. The restricted mixing failed with mask $\{1\}$, $\{1, 2\}$ since the patterns are inferior to the original pattern 00101 in the trap problems. After performing the bit-flipping with the second mask $\{1, 2\}$, the resulting pattern 11101 exists in the population (P_3). Given the original stop criteria, the procedure should continue. This ends up with three more failure recombinations, L_3, L_4, and L_5. These attemps are unnecessary, since they can be performed on the existing chromosome P_3 without wasting the first and second attempts. From another perspective, the following back mixing operators drastically reduce the diversity. So it is important to avoid common trials in restricted mixing to preserve population diversity.

To prevent this situation, I stop the restricted mixing whenever the new generated chromosome already exists in the population. Another receiver along with another set of customized models is chosen for the next iteration of the restricted mixing. This criterion preserves diversity in the population and also reduces NFE waste. The slightly improvement for DSMGA-II is shown in Table 5.1. The pseudo code of the modified restricted mixing is given in Algorithm6.

Problems	**Without Early Stop**	**With Early Stop**	**Ratio**
Concat. trap, $\ell = 400$	54.7	53.6	2.0%
Cyclic trap, $\ell = 400$	116.4	113.1	2.8%
Folded trap, $\ell = 480$	240.4	243.6	-1.3%
NK-S1, $\ell = 400$	723.0	713.3	1.3%
NK-S3, $\ell = 400$	548.2	533.5	2.7%
NK-S5, $\ell = 400$	59.8	59.5	0.5%
2D spin-glass, $\ell = 784$	709.9	683.2	3.8%
MAX-SAT, $\ell = 200$	6346.0	6291.2	0.9%

Table 5.1: Required NFE of DSMGA-II for the largest test problems (unit: K)

doi:10.6342/NTU201704045

Algorithm 6: Modified Restricted Mixing with IsInP

ILS_one: one-edge incremental linkage set,
ILS_two: two-edge incremental linkage set,
f: evaluation function,
P: population, ℓ: problem size,
T: trial solution, M: mask,
R_M: pattern of R extracted by M,
$\overline{R_M}$: complement pattern of R_M
input : R: receiver
output: R: reciever, M: mask

```
for i ← 1 to |ILS_one| do
    M ← ILS_one_i
    if complement(R_M) ⊄ P then
        SupplyBound ← i

for i ← 1 to |ILS_one| do
    M ← ILS_two_i
    if complement(R_M) ⊂ P then
        T ← R
        T_M ← complement(R_M)
        if T ∈ P then
            return (R, ∅)
        if f(T) ≥ f(R) then
            R ← T
            return (R, M)
return (R, ∅)
```


doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Chapter 6

Experiment Results

In this chapter, the experiment setup is described first. The results and discussions are then detailed.

6.1 Experiment Setup

DSMGA-II is an enhanced edition of DSMGA and it outperforms multiple variants of its predecessor [28], hBOA and LT-GOMEA in all benchmark problems. Thus, I will not discuss these algorithms in the following comparison. Here, I focus on comparing the modified DSMGA-II with the original DSMGA-II.

A minimum population is required for a certain number of consecutive successful runs. In the following experiments, I adopt an adaptive sweeping procedure described in [13] for more accurate search of minimum NFE. The sweeping procedure starts with a reasonable population size. If the algorithm cannot consecutively reach the global optimum with the given population, the population size increases with a predefined step. A successful *trial* is defined as 10 consecutive successful hits. The mean of NFE over 10 successful runs is recorded as the minimum evaluations required for a given population size. After a successful trial is recorded, a smaller step is adopted to narrow down the sweeping range. The sweeping procedure continues until the step converges.

The sweeping procedure gives better precision than the canonical bisection procedure, since the acquired resolution increases as the steps decreases after each successful trial.

doi:10.6342/NTU201704045

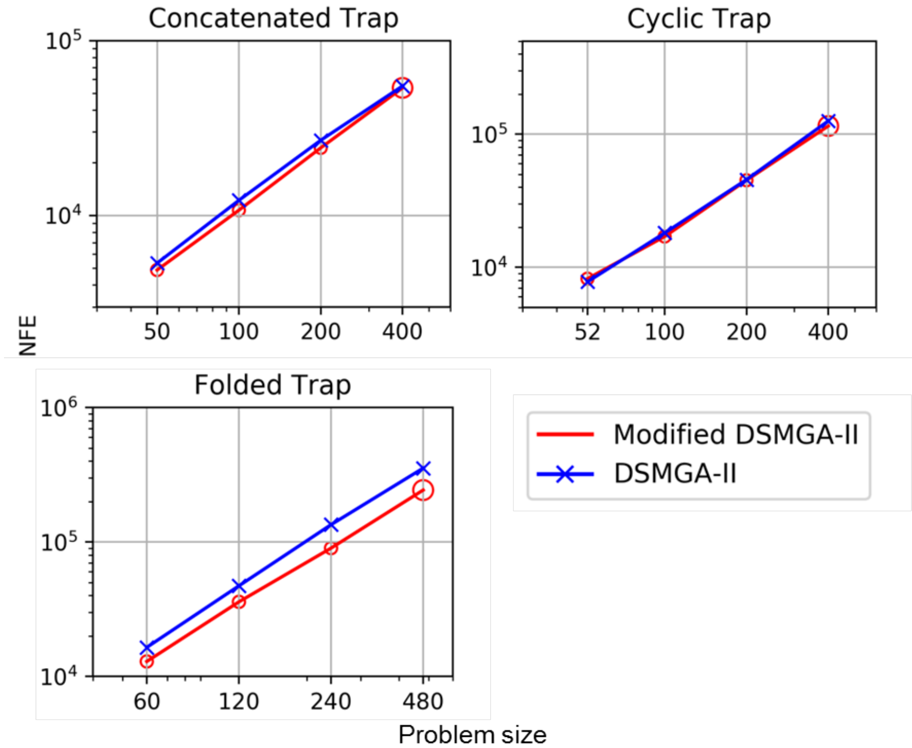


Figure 6.1: Scalability of the modified and the original DSMGA-II on the problems of deceptive variants.

The minimum evaluations required for each problem are averaged over 100 independent runs or instances.

I compare the both versions of DSMGA-II under the following settings. The selection pressure is set as 2. The constant R in Algorithm 1 is set to be $\ell/50$, as suggested in [13].

6.2 Results and Discussions

Comparison on deceptive variants

I first consider the deceptive problems. The results are shown in Figure 6.1. For the concatenated trap problem, there is little difference between the modified DSMGA-II and the original. It is because the problem structure is relatively easy for DSMGA-II to learn. Both subsolutions, 00000 and 11111, are clearly identified in the early stage. As a result, the edge $L(01 \cup 10)$ is rarely used.

For the cyclic trap problem, the two-edge graph helps the linkage model grow toward the correct direction. For instance, consider the pattern "11111**0**000", and the ILS

doi:10.6342/NTU201704045

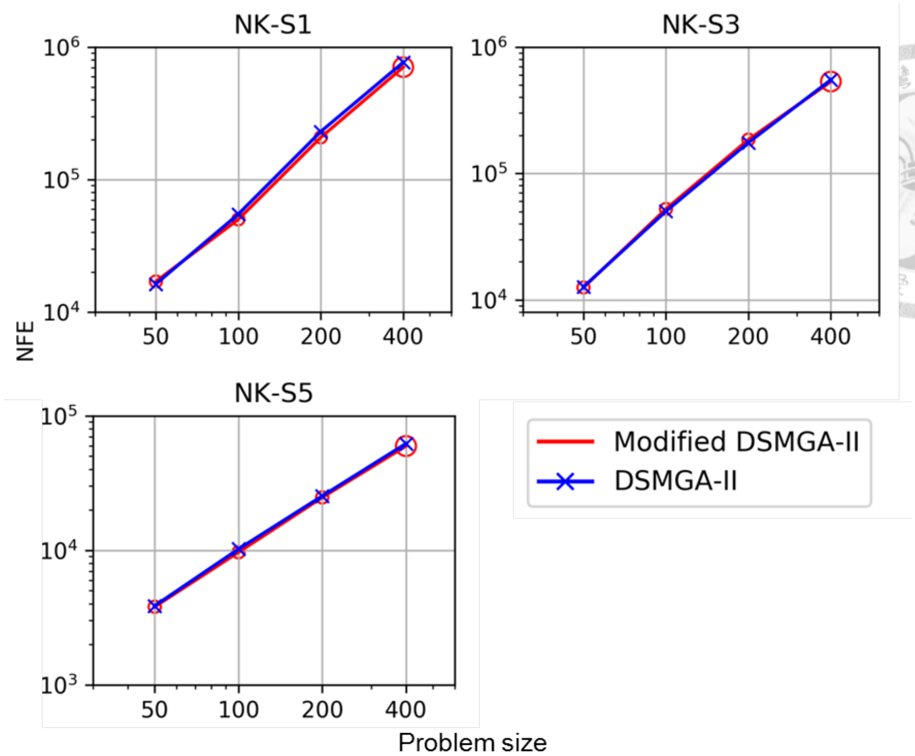


Figure 6.2: Scalability of the modified and the original DSMGA-II on NK-landscape with various degrees of overlapping.

construction starts from the fifth bit which is marked. The two-edge graph suggests the linkage mask to connect its right neighbor instead of its left neighbor since $L(01 \cup 10)$ is much weaker than $L(00 \cup 11)$ for the cyclic trap function. Therefore, a moderate evaluation reduction is achieved.

For the folded trap problem, the two-edge model helps to align the receiver with the prominent pattern in population. The modified version more likely flips the plateau (such as 111000) to the correct subsolutions (000000 or 111111) than DSMGA-II with the same DSM. As a result, significant improvement can be produced.

Comparison on the NK-landscape problems

The results of the NK series in Figure 6.2 demonstrate our modifications do not improve DSMGA-II by much since the NK problems do not have clear structures to for two-edge graph to exploit. The difference slightly enlarges as the degree of overlapping decreases. Nevertheless, our modifications still maintain the performance of the original algorithm.

doi:10.6342/NTU201704045

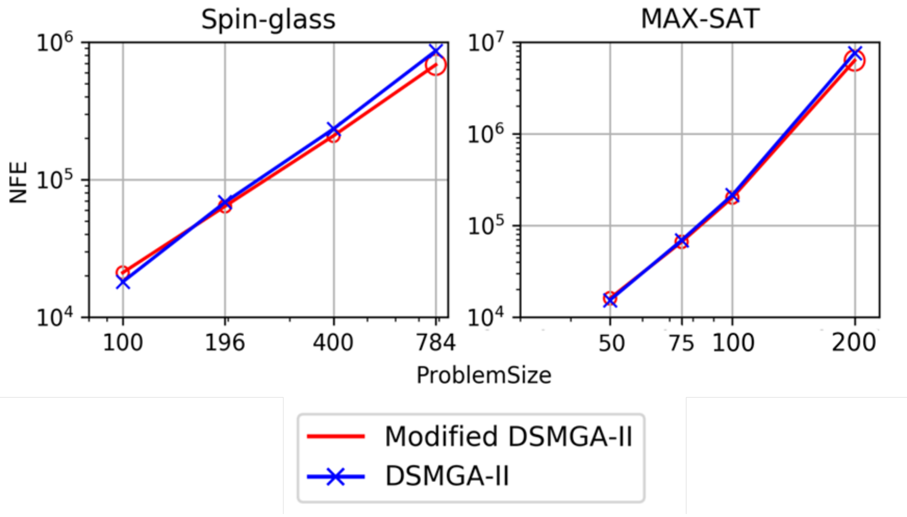


Figure 6.3: Scalability of the modified and the original DSMGA-II on the real world problems.

Comparison on Ising spin-glass and MAX-SAT

For the Ising spin-glass problem in Figure 6.3, the slope of the modified DSMGA-II decreases as the problem size increases. This is due to the numerous numbers of plateaus in the spin-glass problem. Noticed that the smallest instance in the Ising spin-glass, the original DSMGA-II shows superior performance to the modified version. The reason is that the modified version requires a bit larger population size than the original DSMGA-II, and for small instances, GHC costs much more NFEs than the mixing. As for the MAX-SAT problem, 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 evaluations among all the algorithms. To sum up, the modified version yields fewer evaluations and better scalability on both problems.

Comparison with the original DSMGA-II

Some of the improvements may not easily be observed from the above figures due to the logarithmic scale. Here we list the NFEs required to solve the largest problems in Table 6.1. Compared with the original DSMGA-II, the modified version shows significant improvement on five out of eight benchmark problems. The NFE reduction reaches up to 31.3% when dealing with the folded trap problem. The modified version also reduces

doi:10.6342/NTU201704045

over 20% of evaluations on MAX-SAT and 2D spin-glass, two real-world problems.

The two-edge graph constructs customized models for each chromosome. This characteristic shows advantage when facing problems with plateaus. Since the two-edge model tends to align with the most prominent pattern in the population, fewer evaluations are needed to jump out of plateaus due to stronger drifting effect. 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.

In this subsection, I also compare the CPU time. The model building time complexity of the original version of DSMGA-II, which is the summation of the time complexity of building DSM and ILS, is $O(n\ell^2) + O(n\ell^2) = O(n\ell^2)$. For the modified version of DSMGA-II, although the time complexity of model building is also $O(n\ell^2) + O(n\ell^2) = O(n\ell^2)$, the coefficients of both versions are different. The original scheme only builds one DSM during the model building, but the two-edge scheme builds an extra DSM since the two-edge scheme keeps two models during the model building. Besides, the two-edge scheme builds an extra ILS for the same reason. The results of CPU Time are shown in Figure 6.4, 6.6 and 6.8. The results show that although, for the time complexity of model building, the coefficient of the modified DSMGA-II is larger than the coefficient of the original DSMGA-II, the modified DSMGA-II maintains the performance of the CPU time on most of the benchmark problems. The reason is that the modified DSMGA-II requires a less generations (Figure 6.5, 6.7, and 6.9) since the two-edge model constructs the customized recombination mask which enhances the efficiency in the mixing stages.

doi:10.6342/NTU201704045

Problem	ℓ	Original M	SD	Modified M	SD	t	Sig.	Improvement
Concat. trap	400	55.0	7.9	53.6	6.9	1.22	.224	2.52%
Cyclic trap	400	123.2	18.8	113.2	13.7	4.34	.000	8.13%
Folded trap	480	354.6	11.4	243.7	6.0	105.18	.000	31.27%
NKs1	400	767.4	418.0	713.7	376.5	1.74	.081	7.00%
NKs3	400	550.6	210.8	533.5	230.0	1.27	.204	3.10%
NKs5	400	62.1	6.3	59.5	5.1	3.44	.001	4.08%
MAX-SAT	200	7887.6	12470.6	6291.3	10689.5	2.09	.039	20.24%
2D spin-glass	784	863.2	505.0	683.3	321.4	7.95	.000	20.85%

Note. ℓ = Problem Size. M = Mean. SD = Standard Deviation. t = paired t-test statistic.
Sig. with 2-tailed. Improvement = $1-(M_{Modified}/M_{Original})$.

Table 6.1: Required NFE of DSMGA-II for the largest test problems (unit : K evaluations)

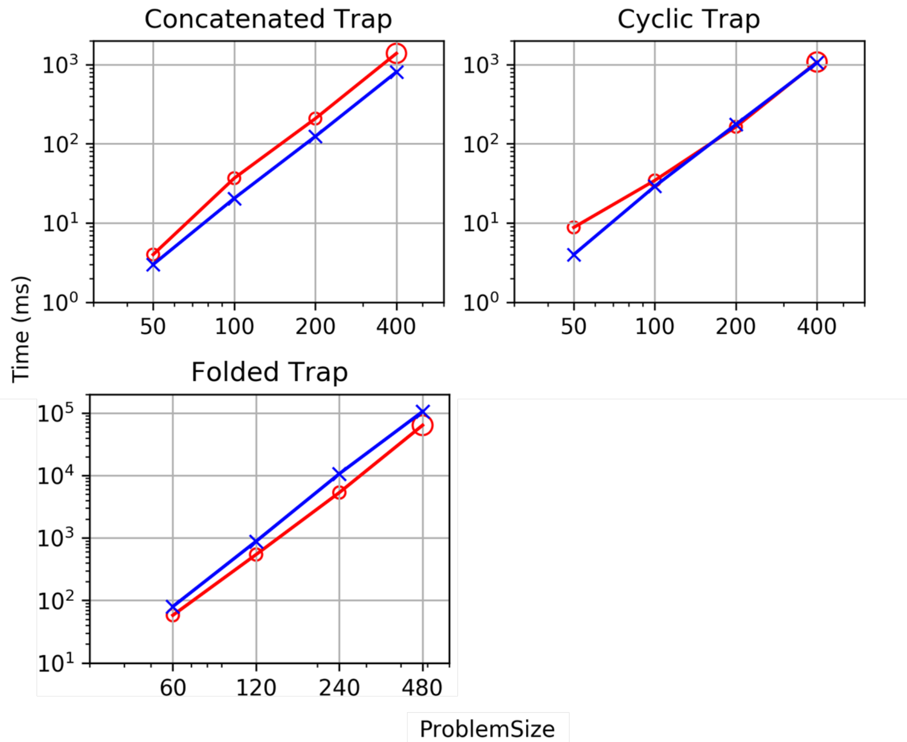


Figure 6.4: CPU time of the modified and the original DSMGA-II on problems of deceptive variants.

doi:10.6342/NTU201704045

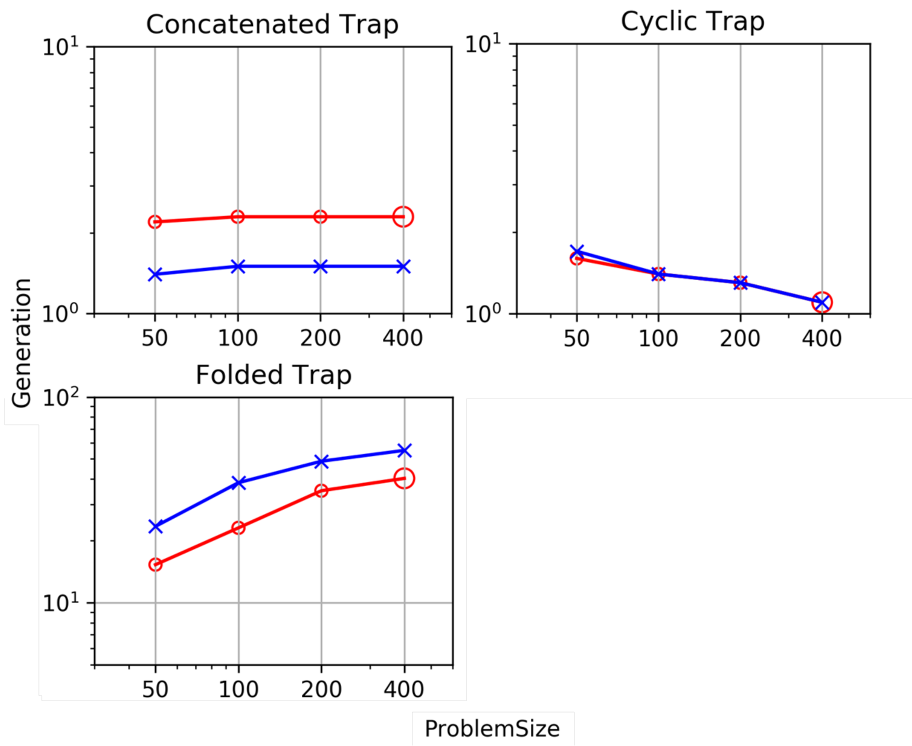


Figure 6.5: Generations of the modified and the original DSMGA-II on problems of deceptive variants.

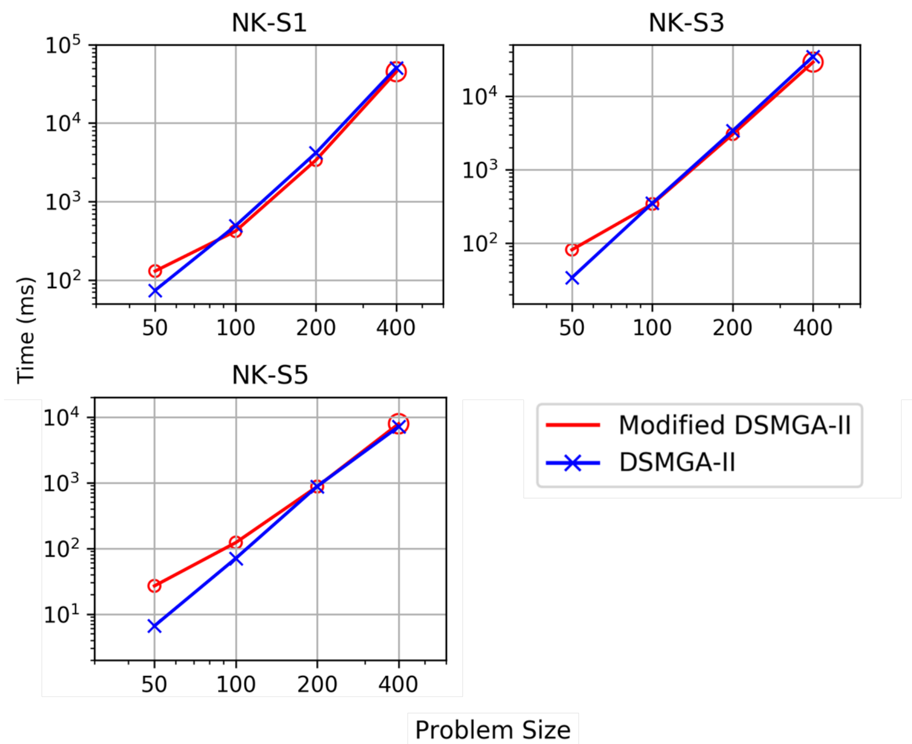


Figure 6.6: CPU time of the modified and the original DSMGA-II on NK-landscape with various degrees of overlapping.

doi:10.6342/NTU201704045

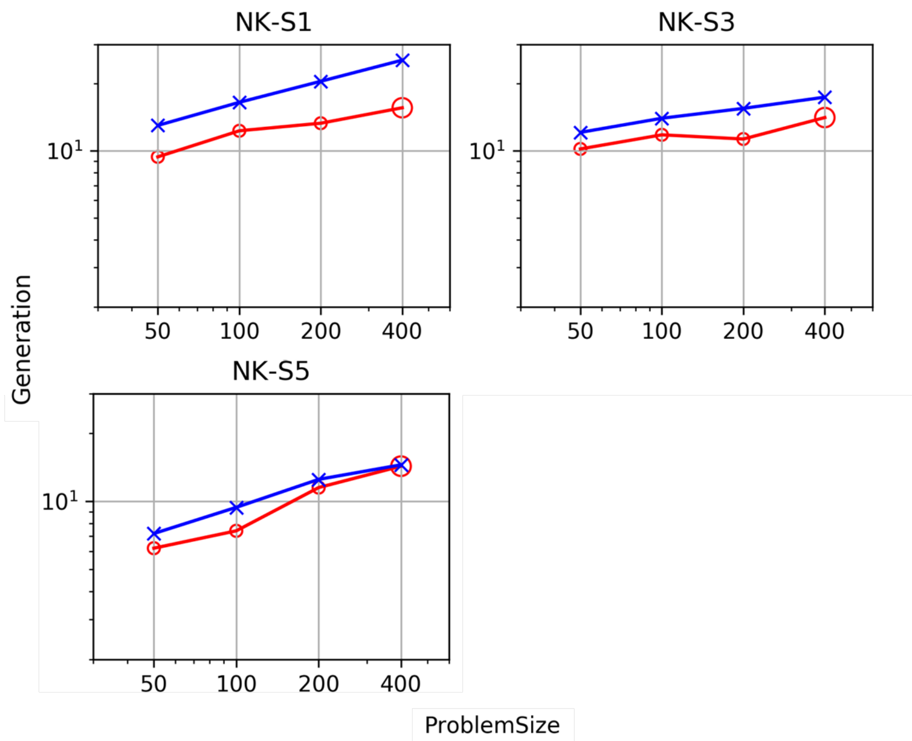


Figure 6.7: Generations of the modified and the original DSMGA-II on NK-landscape with various degrees of overlapping.

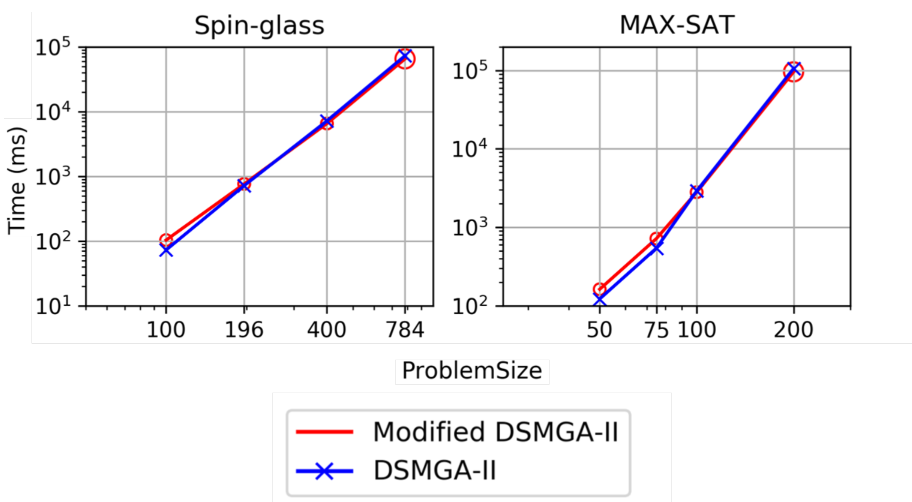


Figure 6.8: CPU time of the modified and the original DSMGA-II on the real world problems.

doi:10.6342/NTU201704045

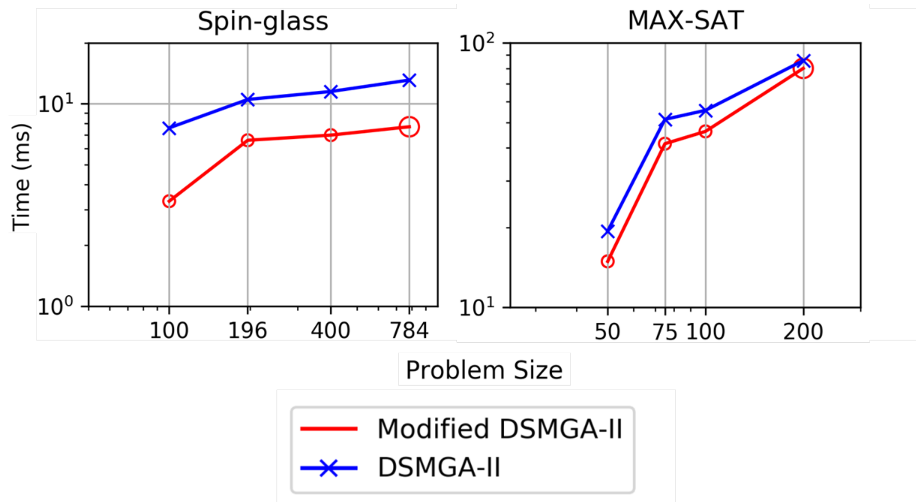


Figure 6.9: Generations of the modified and the original DSMGA-II on the real world problems.

doi:10.6342/NTU201704045

doi:10.6342/NTU201704045

Chapter 7

Conclusion and Future Works

In this thesis, I proposed a customized linkage model called the two-edge linkage model to further improve the performance of the DSMGA-II. In order to achieve this target, I first divide the mutual information into two parts - $L(00 \cup 11)$ and $L(10 \cup 01)$. The reason for such division is to protect the pattern within each building blocks. Then, in the restricted mixing, the new model eliminates the conflict edges according to the alleles of the receiver. Finally, the model customizes the mixing mask for the receiver. The two-edge linkage model provides customized recombination masks and far more possible linkage combinations for every single chromosome. I believe this technique can enhance the efficiency in mixing.

However, the results are unexpectedly frustrating in the beginning. The new issue caused by the customized model is called supply overfitting. When I divide the mutual information into the two complementary groups, the relationship between the strength of the linkage and the length of the supply becomes a ”Chicken-egg” problem, which means the strong linkages imply the long supply and vice versa. This situation increases unnecessary trials in the back mixing since the long pattern is unlikely to be accepted by the population especially in the early stage since the information for building more accurate model is not enough.

To deal with this problem, I first provide a method called the representative supply check, which is a modified version of the supply check in the original DSMGA-II. The concept is that if the population contains a specific ratio of a pattern, the pattern is more

doi:10.6342/NTU201704045

likely to be accepted by other chromosomes in the back mixing. However, the modified supply check just slightly shorten the length of the supply. What worse, in some deceptive case, this scheme causes a disastrous result due to the cross competition.

I then go back to the original supply check. I speculate that the original supply check causes a shorter supply length because of adopting the undivided mutual information. The whole mutual information which is a more global view of the population may provide a supply length which enhance the efficiency in the mixing stages. So I try to adopt this concept in the two-edge linkage model, and this scheme finally works well.

Combined with the other minor modification, the early stop, which reduces unnecessary trials in a specific circumstance in the mixing, the new linkage model shows a stable scalability and improves the original DSMGA-II by 12.2% on average, with the range between 2.5% and 31.3% in terms of NFE. Besides, the modified DSMGA-II maintains the performance of CPU time since the customized model requires less generations.

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

On the other hand, the supply bound should also be further investigated. But before finding another bound which further improves the performance of the DSMGA-II, we should comprehend the concept behind this technique. Just as the two-edge linkage model for the DSMGA-II is based on the concept of customizing, what is the meaning behind the supply bound. Only if we clearly understand the concept behind these methods, we can combine these technique cleverly. Still, further investigations from the theoretical aspect, are needed as well.

doi:10.6342/NTU201704045

Bibliography

[1] P. A. Bosman and D. Thierens. Linkage neighbors, optimal mixing and forced improvements in genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 585–592, 2012.

[2] F. Chicano, D. Whitley, G. Ochoa, and R. Tinós. Optimizing one million variable nk landscapes by hybridizing deterministic recombination and local search. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 753–760, 2017.

[3] L. Davis. Applying adaptive algorithms to epistatic domains. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 1, pages 162–164, 1985.

[4] 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.

[5] K.-C. Fan, T.-L. Yu, and J.-T. Lee. Linkage learning by number of function evaluations estimation: Practical view of building blocks. *Information Sciences*, 230:162–182, 2013.

[6] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1 edition, 1989.

[7] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

doi:10.6342/NTU201704045

[8] D. E. Goldberg and R. Lingle. Alleles, loci, and the traveling salesman problem. In *Proceedings of the International Conference on Genetic Algorithms*, pages 154–159, 1985.

[9] D. E. Goldberg, K. Sastry, and T. Latoza. On the supply of building blocks. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 336–342, 2001.

[10] J. Grefenstette, R. Gopal, B. Rosmaita, and D. Van Gucht. Genetic algorithms for the traveling salesman problem. In *Proceedings of the International Conference on Genetic Algorithms*, pages 160–168, 1985.

[11] G. Harik. Linkage learning via probabilistic modeling in the ecga. Technical report, University of Illinois, 1999.

[12] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.

[13] 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 Genetic and Evolutionary Computation Conference*, pages 519–526, 2015.

[14] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[15] P. Larranaga. A review on estimation of distribution algorithms. In *Estimation of distribution algorithms*, pages 57–100. Springer, 2002.

[16] I. Oliver, D. Smith, and J. R. Holland. Study of permutation crossover operators on the traveling salesman problem. In *Proceedings of the International Conference on Genetic Algorithms*, 1987.

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

doi:10.6342/NTU201704045

[18] M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21(1):5–20, 2002.

[19] M. Pelikan, M. W. Hauschild, and D. Thierens. Pairwise and problem-specific distance metrics in the linkage tree genetic algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1005–1012, 2011.

[20] M. Pelikan, K. Sastry, and E. Cantú-Paz. *Scalable optimization via probabilistic modeling: From algorithms to applications*, volume 33. Springer, 2007.

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

[22] D. Thierens and P. A. Bosman. Optimal mixing evolutionary algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 617–624, 2011.

[23] D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*, pages 38–47, 1993.

[24] R. L. R. Thomas H. Cormen, Charles E. Leiserson and C. Stein. *Introduction to Algorithms*. MIT Press, 2 edition, 1992.

[25] R. Tintos, D. Whitley, and F. Chicano. Partition crossover for pseudo-boolean optimization. In *Proceedings of the ACM Conference on Foundations of Genetic Algorithms XIII*, pages 137–149, 2015.

[26] M. Tsuji, M. Munetomo, and K. Akama. A crossover for complex building blocks overlapping. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1337–1344, 2006.

[27] L. D. Whitley, T. Starkweather, and D. Fuquay. Scheduling problems and traveling salesmen: The genetic edge recombination operator. In *Proceedings of the International Conference on Genetic Algorithms*, pages 133–140, 1989.

doi:10.6342/NTU201704045

[28] 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 *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1620–1621, 2003.

[29] T.-L. Yu, K. Sastry, and D. E. Goldberg. Linkage learning, overlapping building blocks, and systematic strategy for scalable recombination. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1217–1224, 2005.

[30] 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 Genetic and Evolutionary Computation Conference*, pages 601–608, 2007.

doi:10.6342/NTU201704045