

parallel data. That is, Speakers A and B must read hundreds of sentences with the same content to teach the machine how to transform content between their voices, which is not practical. A new GAN-based approach to VC is proposed. Only audio content from Speakers A and B is needed. They do not have to read the same sentences, and they do not even have to speak the same language.

Unsupervised speech recognition [4,5]

A completely unsupervised speech recognition framework in which only unrelated speech utterances and text sentences are needed for model training is proposed. An unsupervised phoneme recognition accuracy of 36% is achieved in the preliminary experiments. This is the first

attempt at reaching the goal of completely unsupervised speech recognition. With this technology, a machine can learn a new language in a novel linguistic environment with little supervision. Imagine an intelligent assistant bought by a family speaking Taiwanese. Although it does not understand Taiwanese at the beginning, by hearing people speaking Taiwanese, it automatically learns the new language.

References

1. Wang, Y., and Lee, H. (2018). Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks. Submitted to *EMNLP*, 2018.
2. Lee, C., Wang, Y., Hsu, T., Chen, K., Lee, H., and Lee, L. (2018). Scalable Sentiment for Sequence-to-sequence Chatbot Response with Performance Analysis. *ICASSP*, 2018.
3. Chou, J., Yeh, C., Lee, H., and Lee, L. (2018). Multi-target Voice Conversion without Parallel Data by Adversarially Learning Disentangled Audio Representations. *INTER-SPEECH*, 2018.
4. Liu, D., Chen, K., Lee, H., and Lee, L. (2018). Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings, *INTERSPEECH*, 2018.
5. Chen, Y., Shen, C., Huang, S., and Lee, H. (2018). Towards Unsupervised Automatic Speech Recognition Trained by Unaligned Speech and Text only. *arXiv:1803.10952* [cs.CL]

Hung-yi Lee

Assistant Professor, Department of Electrical Engineering
hungyilee@ntu.edu.tw

Dependency structure matrix genetic algorithm II

Introduction

In 2015, we proposed the dependency structure matrix genetic algorithm II (DSMGA-II) [1], which is currently a state-of-the-art discrete genetic algorithm. 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 major recombination operators of DSMGA-II. Such a combination significantly reduces the number

of function evaluations (NFE) compared with those of 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) [2] and the hierarchical Bayesian optimization algorithm (hBOA) [3], two previously used state-of-the-art genetic algorithms, on various benchmark problems.

DSMGA-II consists of four major components: pairwise linkage detection, model build-

ing, restricted mixing and back mixing. First, DSMGA-II randomly initializes a population. Then, to enhance the quality of the pairwise linkage model and reduce noise in the population, DSMGA-II performs bit-flipping greedy hill climbing (GHC) on each chromosome. Performing GHC before linkage model building can further ensure pairwise linkage information between genes by ruling out trivial cases that can be solved without linkage information. After initializing the population, only the selected chromosomes are used to build

the linkage model. DSMGA-II adopts mutual information as a pairwise linkage measure and stores linkage information in a DSM. Then, recombination via the restricted and back mixings

proceeds with the original population. To prevent overfitting due to frequent model building, the algorithm updates DSM once every $O(l)$ generations, where l is the problem size. Note that the

population after selection is only used for updating the DSM; the rest of the algorithm proceeds with the original population. The pseudo code of DSMGA-II is presented in Algorithms 1, 2, and 3.

Algorithm 1: DSMGA-II

\mathcal{P} : population, ℓ : problem size \mathcal{D} : DSM, \mathcal{L} : incremental linkage set, R : constant

randomly initialize population \mathcal{P}

$\mathcal{P} \leftarrow \text{RUNLOCALSEARCH}(\mathcal{P})$

while $\neg \text{SHOULDTERMINATE}$ **do**

$\mathcal{S} \leftarrow \text{TOURNAMENTSELECTION}(\mathcal{P}, s)$

$\mathcal{D} \leftarrow \text{UPDITEMATRIX}(\mathcal{S})$

for $k = 1$ **to** R **do**

$\mathcal{I} \leftarrow$ random permutation from 1 to $|\mathcal{P}|$

for $i = 1$ **to** $|\mathcal{P}|$ **do**

$(P_{I_i}, L_s) \leftarrow \text{RESTRICTEDMIXING}(P_{I_i})$

$\mathcal{P} \leftarrow \text{BACKMIXING}(P_{I_i}, L_s)$

return the best instance in \mathcal{P}

Algorithm 2: Restricted Mixing

\mathcal{P} : population, ℓ : problem size, C : set of vertices in AMWCS \mathcal{L} : incremental linkage set, f : evaluation function, T : trial solution

Input: P : receiver

AMWCS \leftarrow random number from 1 to ℓ

while $P'_C \in \mathcal{P}$ **do**

$\mathcal{L} \leftarrow \mathcal{L} \cup C$

 join the nearest vertex into AMWCS

for $i = 1$ **to** $|\mathcal{L}|$ **do**

$T \leftarrow P$

$T_{L_i} \leftarrow T'_{L_i}$

if $f(T) \geq f(P)$ **and** $T \notin \mathcal{P}$ **then**

$P \leftarrow T$

return (P, L_i)

Algorithm 3: Back Mixing

\mathcal{P} : population, f : evaluation function, T : trial solution E : set of candidate solutions

Input: D : donor, L : mask

$\text{improved} \leftarrow \text{false}$

for $j = 1$ **to** $|\mathcal{P}|$ **do**

$T \leftarrow P_j$

$T_L \leftarrow D_L$

if $f(T) > f(P_j)$ **then**

$P_j \leftarrow T$

$\text{improved} \leftarrow \text{true}$

else

if $f(T) = f(P_j)$ **then**

$E \leftarrow E \cup \{T\}$

if $\neg \text{improved}$ **then**

accept all solutions in E

return \mathcal{P}

Performance of DSMGA-II

Six types of linkage benchmark problems are tested, including four classical linkage-underlying problems and two real-world problems. These benchmark problems cover the wide range of different characteristics of real-world problems. Their mathematical formulations are listed in the following table.

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

Table 1. Six types of linkage benchmark problems are tested, including four classical linkage-underlying problems and two real-world problems.

The experimental results are shown in Figures 1 to 3. Figure 1 shows that the differences between DSMGA-II and the other algorithms become larger as the degree of overlap decreases. During the initial back mixing process, a few function evaluations are performed to refine the

graph. As the linkage information becomes clearer, the correct subsolution stands out, and back mixing causes the subsolution to quickly dominate the population. This approach makes the algorithm more efficient. However, if the problem structures severely overlap, the graph refining pro-

cess is prolonged, and hence, the NFE increases. The results indicate that DSMGA-II is capable of handling problems with overlapping structures, even for randomly generated problem landscapes. In other words, the ILS model expresses overlapping relations well.

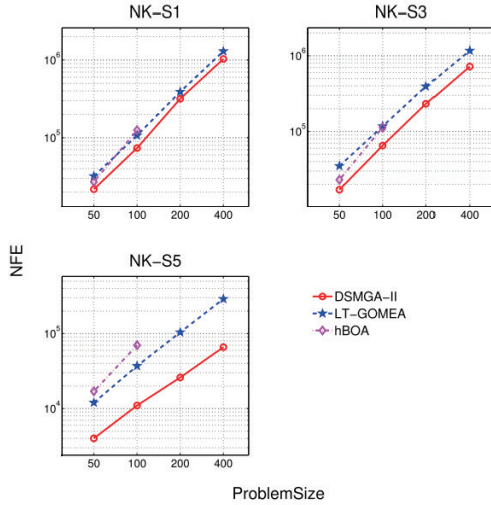


Figure 1. Scalability of DSMGA-II, LT-GOMEA and hBOA on NK-landscape problems with various degrees of overlap.

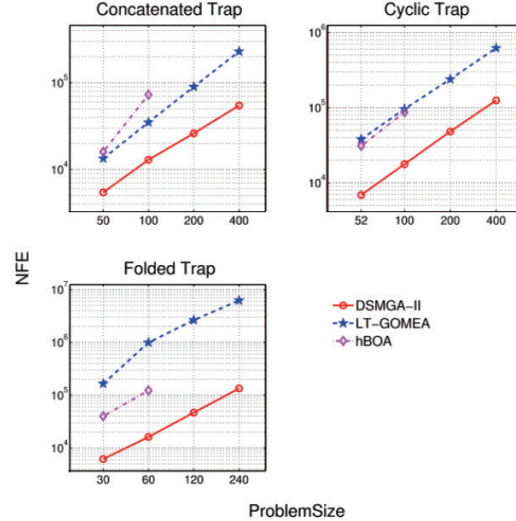


Figure 2. Scalability of DSMGA-II, LT-GOMEA and hBOA on the problems of deceptive variants.

The results pertaining to deceptive variants are shown in Figure 2. For the concatenated trap, the slope of DSMGA-II decreases as the problem becomes larger. For the cyclic trap, the scalabilities of all algorithms appear very similar to that of DSMGA-II, which is the lowest by a constant factor. As previously mentioned, the cyclic trap cannot be solved efficiently merely by the correct problem decomposition. The use

of linkage information is also key. ILS automatically extends the mask for trials and stops on the first successful recombination to avoid spending unnecessary function evaluations, while many other genetic algorithms utilize the linkage models determined by certain thresholds, which are sensitive to the parameter settings. The folded trap contains a large number of local optima that reside on plateaus. Performing

an efficient search without losing too much diversity is the key to conquering such a difficulty. Back mixing leads to the drift of subsolutions as necessary, and restricted mixing checks whether the trial solution is unique in the population to maintain diversity. As a result, DSMGA-II shows a good ability to address attractions without trading efficiency for diversity.

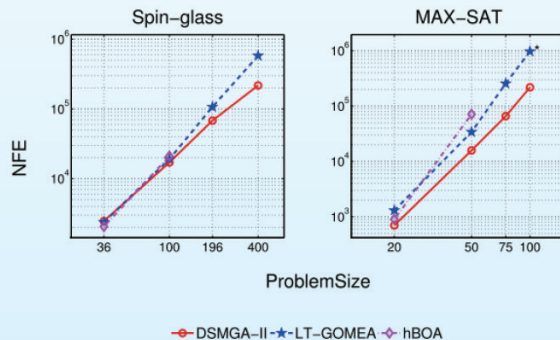


Figure 3. Scalability of DSMGA-II, LT-GOMEA and hBOA on Spin-glass and MAX-SAT (*LT-GOMEA fails to reach the global optima for two instances with $l = 100$ on MAX-SAT).

For the Ising spin-glass problems, the slope of DSMGA-II decreases as the problem becomes larger (Figure 3). This trend is probably observed because the problems contain a large number of plateaus, for which the advantages of DSMGA-II hold. The overall amount of time spent on fitness evaluations appears to grow polynomially as $O(n^3)$ for DSMGA-II, which is close to the best known result of problem-specific algorithms for Ising spin-glass problems. For MAX-

SAT problems, although the NFE of DSMGA-II still grows exponentially (because it is a well-known NP-complete problem), DSMGA-II consumes fewer functional evaluations than hBOA and LT-GOMEA.

In 2017, we proposed modifying the linkage model [4] by separating two different types of linkage and produced a two-edge graphical model (Figure 4). With this modified linkage model, DSMGA-II consumes 5~30%

fewer function evaluations on the abovementioned test problems.

In summary, genetic algorithms have been widely used for black-box optimization in many different areas of engineering. Having a more competent optimizer means that we can have a better performing system, a stronger architecture, a faster bullet train (Figure 5), and most importantly, a more convenient life.

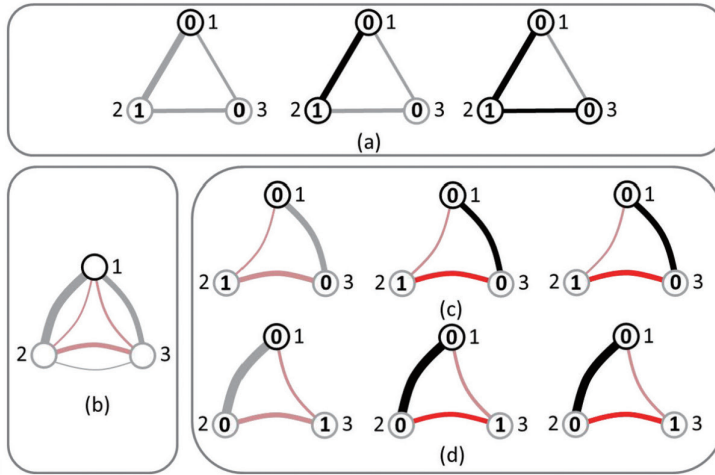


Figure 4. (a) The original linkage model in DSMGA-II. (b) The two-edge graphical model, where the homogenous linkages are black and the heterogeneous linkages are red. (c) and (d) Two cases of ILS for different alleles.



Figure 5. The shape of the nose cone of the N700 bullet train was designed by genetic algorithms.

References

1. Hsu, S.-H and Yu, T.-L. (2015). Optimization by pairwise linkage detection, incremental linkage set, and restricted/back mixing: DSMGA-II. *Annual Conference on Genetic and Evolutionary Computation*, 519–526.
2. Thierens, D. and Bosman, P. A. (2011). Optimal mixing evolutionary

algorithms. In *Proceedings of the 13th annual conference on Genetic and Evolutionary Computation Conference*, 617–624.

3. Pelikan, M. and Goldberg, D. E. (2001). Escaping hierarchical traps with competent genetic algorithms. In *Proceedings of Genetic and Evolutionary Computation Conference*, 511–518.

4. Chen, P.-L., Peng, C.-J., & Yu, T.-L. (2017). Two-edge graphical linkage model for DSMGA-II. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 745–752.

Tian-Li Yu

Associate Professor, Department of Electrical Engineering
 tianliyu@ntu.edu.tw