Introducing a Linkage Identification Considering non-Monotonicity to Multi-objective Evolutionary Optimization with Decomposition for Real-valued Functions

Kousuke Izumiya
Graduate School of Information Science and Technology,
Hokkaido University
Sapporo, Japan
sunlightheart@eis.hokudai.ac.jp

Masaharu Munetomo
Information Initiative Center,
Hokkaido University
Sapporo, Japan
munetomo@iic.hokudai.ac.jp

ABSTRACT

In this paper, we propose MOEA/D-LIEM^2 to optimize multiple real-valued objective functions which have complex interaction among variables by employing linkage identification. We compared the proposed algorithm with MOEA/D without linkage identification. As a result, we found that the proposed algorithm outperforms the original MOEA/D for problems consisting of two deceptive functions with linkage.

CCS CONCEPTS

• **Theory of computation → Evolutionary algorithms**;

KEYWORDS

Multi-objective optimization, Linkage identification, Real-valued function

ACM Reference Format:
Kousuke Izumiya and Masaharu Munetomo. 2018. Introducing a Linkage Identification Considering non-Monotonicity to Multi-objective Evolutionary Optimization with Decomposition for Real-valued Functions. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3205651.3205692

1 INTRODUCTION

In real-world problems, it is necessary to optimize objective functions that have complex interaction among variables, which is called *linkage*. A series of algorithms has been proposed to identify linkage in genetic algorithms such as LINC, LIMD[4], LIEM^2[5] etc. They can identify linkage groups for binary-coded fitness functions by employing bit-wise perturbations. For real-valued objective functions, some algorithms such as LINC-R or LIDI-R have been proposed to detect linkage for real variables. For multi objective optimization problems, MOEA/D-LIMD[3] has been proposed to detect linkage information for binary-coded multi-objective optimization problems. In this paper, we propose MOEA/D-LIEM^2 to detect linkage groups based on non-monotonicity detection for real-value multi-objective optimization problems. The proposed algorithm aims at improving the quality of solutions to complex multi-objective optimization problems by incorporating linkage identification technique LIEM^2 which calculates epistasis measures considering non-monotonicity caused by pair-wise perturbations.

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205692

2 PROPOSED ALGORITHM

Several real-coded GAs have been studied to deal with objective functions of real variables. In this proposed approach, we employ UNDX (Unimodal Normal Distribution Crossover) [1] and a linkage identification technique called LIEM^2 (Linkage Identification with Epistasis Measures considering non-Monotonicity) for solving real-valued functions with linkage. LIEM^2 detects non monotonicity of the difference in fitness values caused by pair-wise perturbations of variables. In LIEM^2, epistasis measure $e_{i,j}$ is defined as follows:

$$e_{i,j} = \max_{s \in P} g(\Delta f_{i,j}(s), \Delta f_i(s), \Delta f_j(s)) \tag{1}$$

$$\Delta f_i(s) = f(\ldots, s_i + \Delta s_i, \ldots) - f(\ldots, s_i, \ldots) \tag{2}$$

$$\Delta f_j(s) = f(\ldots, s_j + \Delta s_j, \ldots) - f(\ldots, s_j, \ldots) \tag{3}$$

$$\Delta f_{i,j}(s) = f(\ldots, s_i + \Delta s_i, \ldots, s_j + \Delta s_j, \ldots) - f(\ldots, s_i, \ldots, s_j, \ldots) \tag{4}$$

$$g(x, y, z) = \begin{cases} tr(y - x) + tr(z - x) & (y > 0, z > 0) \\ tr(x - y) + tr(x - y) & (y < 0, z < 0) \\ 0 & otherwise \end{cases} \tag{5}$$

$$tr(x) = \begin{cases} x & (x > 0) \\ 0 & otherwise \end{cases} \tag{6}$$

After sorting $e_{i,j}$, the combinations of variable(i,j) with higher $e_{i,j}$ value are detected as linkage. As perturbation on real-valued function(in (2),(3),(4)), we apply random changes of the values within the domain for each variable. Unlike binary strings where perturbations only need bit-flips, real-valued functions can be perturbed within their range, thus the number of individuals necessary for linkage identification increases. The number of individuals used for linkage identification this time was experimentally examined as 500. MOEA/D-LIMD[3] has been proposed by combining these methods. We extended it to real-valued function using LIEM^2 instead of LIMD. The algorithm of MOEA/D-LIEM^2 is shown in Algorithm1. Step 1 detects linkages using LIEM^2. Step 2 is the same as MOEA/D. After that, for each linkage, sub-solutions that give an optimal value to the function are searched using UNDX and a specified number of sub-solutions are stored as a candidate

Algorithm 1 MOEA/D-LIEM^2

1: Linkage Identification using LIEM^2
2: Calculate the weight vector and the reference point z for MOEA/D
3: Generate sub-solutions based on linkage
4: Generate initial solutions based on sub-solutions
5: Crossover sub-solutions with function which have high value in weight vector
6: When stopping criteria is satisfied, stop and output solutions

solutions set. Step 4 generates the initial solution set for the main search from candidate solution sets. Crossover operation on identified linkage unit is performed in step 5. We search by linking variables belonging to linkage randomly selected from linkages for function with large weight vector value.

3 EXPERIMENT

In order to evaluate the performance of our method for real-valued function, we compared the proposed method and MOEA/D. We employed two test functions as follows:

$$\min \mathrm{f_{trap}}(\mathrm{x}) = \sum_{\mathrm{k=0}}^{\mathrm{n/5}} \mathrm{trap}(\mathrm{x_{5k}} : \mathrm{x_{5k+4}})$$

$$\min \mathrm{f_{invtrap}}(\mathrm{x}) = \sum_{\mathrm{k=0}}^{\mathrm{n/5}} \mathrm{invtrap}(\mathrm{x_{5k}} : \mathrm{x_{5k+4}})$$

$$trap(x) = \begin{cases} -\sum_k(x_k) & -\sum(x_k) < -4.0 \\ -4.0 + \sum(x_k) & -\sum(x_k) \geq -4.0 \end{cases}$$

$$invtrap(x) = \begin{cases} -5.0 + \sum(x_k) & -\sum(x_k) > -1.0 \\ -\sum(x_k) + 1.0 & -\sum(x_k) \leq -1.0 \end{cases}$$

$$x_k \in [0.1]\ (k = 1, \ldots, n)$$

We evaluated the performance of our method from the final solution sets found by 10 trial runs. We set up the algorithm parameter as in [3]. The experimental results indicated that the longer the length of solution is, the more difficult to find good solutions. In both weightedsum or tchebychev approaches, the results did not change much. In addition, unlike in previous study [3], there was no difference in results as to whether the position of the linkage. For integer-valued functions, it is possible to solve easily by the fact that the positions of the variables included in the same linkage are close to each other, whereas for real-valued functions it is difficult to solve regardless of its position. UNDX or CMA-ES can capture some dependencies of variables found by random mutation; However, when a problem has complex structure, it is insufficient to detect linkages. On the other hand, MOEA/D-LIEM^2 can detect linkages directly, and utilizes those linkages to search optimal solutions, thus better solutions can be found for every conditions. Moreover, solutions obtained by MOEA/D without linkage identification are far from Pareto front because it does not consider interactions among variables. Therefore, the proposed method is effective for solving problems with deceptive functions.

Table 1: settings of experiment

length of solution n	(30, 100)
size of population N	201
size of neighbor solution T	5
stopping criteria	500 generations
tightness of linkage	(shuffled, ordered)[1]
aggregation function	(weightedsum, tchebychev)

[1] in shuffled linkage, locus order is as follows:
n=30:[3, 25, 21, 27, 19, 1, 26, 7, 15, 22, 14, 5, 12, 6, 2, 28, 8, 18, 4, 23, 29, 10, 24, 9, 11, 16, 20, 17, 0, 13]
n=100:[37, 89, 40, 78, 68, 71, 14, 44, 45, 42, 64, 48, 97, 1, 98, 49, 10, 25, 56, 17, 0, 61, 34, 81, 62, 24, 60, 92, 57, 21, 47, 59, 33, 95, 55, 50, 53, 15, 36, 84, 38, 69, 41, 8, 43, 31, 12, 32, 72, 2, 5, 88, 58, 86, 77, 7, 93, 74, 52, 82, 83, 26, 80, 54, 3, 6, 94, 13, 51, 4, 66, 29, 76, 70, 39, 35, 16, 46, 9, 85, 73, 27, 20, 19, 87, 65, 99, 18, 23, 30, 63, 75, 11, 79, 67, 22, 91, 90, 28, 96]

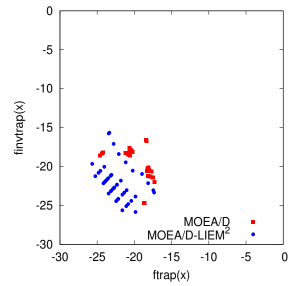

Figure 1: n=30, weightedsum, shuffled

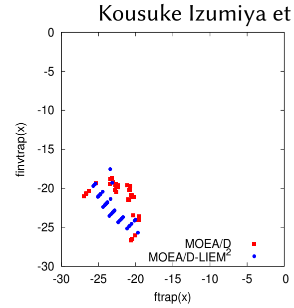

Figure 2: n=30, tchebychev, shuffled

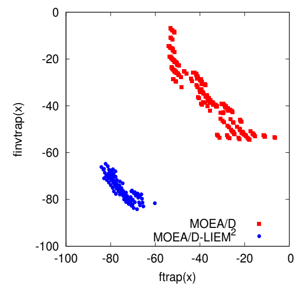

Figure 3: n=100, weighted-sum, shuffled

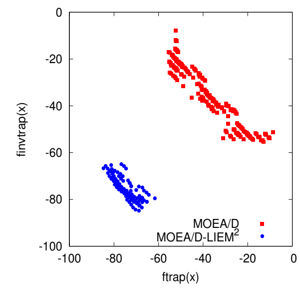

Figure 4: n=100, weighted-sum, ordered

4 CONCLUSION

Optimizing real-valued functions with linkages is challenging in MOP. In this paper, we proposed MOEA/D-LIEM^2 to solve those problems. We evaluated the performance of our method by comparing with the original MOEA/D. The results indicated that our proposed algorithm can find better solutions for real-valued functions which have linkage. As future research, we plan to apply our proposed approach to other MOP functions having different linkages.

REFERENCES

[1] I. Ono, H. Kita, S. Kobayashi. . A real-coded genetic algorithm using the unimodal normal distribution crossover. Advances in evolutionary computing, 213-237. 2003.
[2] Q. Zhang, H. Li. MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition, IEEE Trans. on Evolutionary Computation, 11(6), 712-731, 2007.
[3] K. Izumiya, M. Munetomo. Multi-objective Evolutionary optimization based on Decomposition with Linkage Identification considering monotonicity, In Evolutionary Computation (CEC), 2017 IEEE Congress on (pp. 905-912). IEEE.
[4] M. Munetomo, D.E. Goldberg. Linkage identification by non-monotonicity detection for overlapping functions. Evolutionary Computation, 7(4), 377-398, MIT Press, 1999.
[5] M. Munetomo. Linkage identification with epistasis measure considering monotonicity conditions. Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, 550-554, 2002.