

# Empirical Linkage Learning

Michał W. Przewozniczek  
 Department of Computational Intelligence  
 Wrocław University of Science and Technology  
 Wrocław, Poland  
 michal.przewozniczek@pwr.edu.pl

Marcin M. Komarnicki  
 Department of Computational Intelligence  
 Wrocław University of Science and Technology  
 Wrocław, Poland  
 marcin.komarnicki@pwr.edu.pl

## ABSTRACT

Linkage learning is employed by many state-of-the-art evolutionary methods designed for solving problems in discrete domains. The effectiveness of these methods is dependent on linkage quality. The linkage may suffer to two different inaccuracy types. If some of the gene dependencies are not discovered, then the missing linkage takes place. Oppositely, if linkage identifies some gene dependencies that do not exist, then the false linkage takes place. To the best of our knowledge, all linkage learning techniques proposed that far predict that some genes are dependent and can commit both of the mistake types. Therefore, we propose a more direct approach. We disturb the genotype and check how these disturbances have influenced the effects of local search. We prove that the proposed Linkage Learning based on Local Optimization (3LO) will never report any false linkage, although it may still miss some true gene dependencies. 3LO is fundamentally different from other linkage learning techniques. Its main disadvantage is a high computational cost and it is not suitable for already known evolutionary methods that frequently compute linkage. Therefore, we propose an evolutionary method that employs 3LO. More details considering 3LO, linkage quality and diversity may be found in the original paper [5].

## CCS CONCEPTS

• Computing methodologies → Artificial intelligence;

## KEYWORDS

Linkage Learning, Genetic Algorithm, Parameter-less

## ACM Reference Format:

Michał W. Przewozniczek and Marcin M. Komarnicki. 2020. Empirical Linkage Learning. In *Genetic and Evolutionary Computation Conference Companion (GECCO '20 Companion)*, July 8–12, 2020, Cancún, Mexico. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3377929.3397771>

## 1 LINKAGE LEARNING BASED ON LOCAL OPTIMIZATION (3LO)

To the best of our knowledge, all linkage learning techniques proposed so far are based on predictions. The base of prediction may be a probabilistic model [4], a gene position in the genotype [3],

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).  
 GECCO '20 Companion, July 8–12, 2020, Cancún, Mexico  
 © 2020 Copyright held by the owner/author(s).  
 ACM ISBN 978-1-4503-7127-8/20/07.  
<https://doi.org/10.1145/3377929.3397771>

graphs, or other data structures created with the use of information theory [1, 6]. Nevertheless, Prediction-based Linkage Learning techniques may discover a *false linkage*. Linkage Learning based on Local Optimization (3LO) checks the differences in genotypes resulting from local optimization. First, we optimize the original genotype. Then, we perturb it and optimize the modified one. Finally, we compare the differences. Thus, the above procedure may be interpreted as an empirical check. If 3LO discovers any two genes as a dependent, the prediction is replaced by certainty. 3LO was proposed for binary domains, but it may be extended to other domains as well.

Let us consider a single-objective optimization problem formulated as:

$$\max_{\vec{x} \in X} \{f(\vec{x})\} \quad (1)$$

where solution  $\vec{x} = [x_1, \dots, x_n]$  is a vector of  $n$  binary decision variables and  $X$  is a set of available solutions. We define  $\vec{x}^{(m)}$ :

$$\vec{x}^{(m)} = [x_1, \dots, x_{m-1}, x'_m, x_{m+1}, \dots, x_n] \quad (2)$$

where  $x'_m$  is the negation of  $x_m$ .

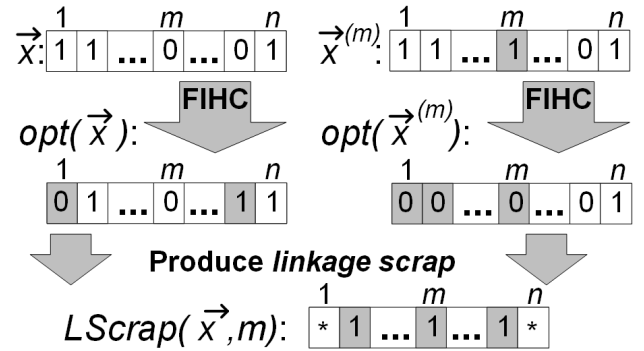


Figure 1: The idea of linkage scrap gathering in 3LO

In 3LO, the basic linkage information is denoted as *linkage scrap* and is defined as follows:

$$LScrap(\vec{x}, m) = OR(XOR(\vec{x}, \vec{x}^{(m)}), XOR(opt(\vec{x}), opt(\vec{x}^{(m)}))) \quad (3)$$

where  $opt(\vec{x})$  is a vector obtained after the optimization of  $\vec{x}$  with a local optimizer. For this purpose we use the First Improvement Hill Climber (FIHC) [1].

The idea of *linkage scrap* gathering is presented in Fig. 1. First, a single gene  $x_m$  of an individual is flipped. The *linkage scrap* informs

**Table 1: Effectiveness comparison for some of the considered test cases**

Problem Class	Problem	n	3LOa			P3			psDSMGA-II			LT-GOMEA		
			Sol ved [%]	Time med. [s]	FFE med.	Sol ved [%]	Time med. [s]	FFE med.	Sol ved [%]	Time med. [s]	FFE med.	Sol ved [%]	Time med. [s]	FFE med.
Dec.	Bimodal 10	2000	100	567	2.5E+7	0	N/A	N/A	83	5 146	9.9E+7	100	13 021	4.0E+8
Conc.	Bim.10 nois.	2000	100	573	2.5E+7	0	N/A	N/A	0	N/A	N/A	0	N/A	N/A
	Dec. 5	2000	100	12 725	7.0E+8	100	158	4.8E+5	100	2 502	1.2E+6	100	126	5.8E+6
	St.Dec. 5	2002	100	8 771	1.2E+8	0	N/A	N/A	97	16 271	8.1E+8	0	N/A	N/A
	Discr. Rast.	800	100	1 293	1.4E+8	100	17	2.0E+5	100	323	5.2E+6	100	12	6.9E+5
Over-lapp.	NK-land.	600	100	4 658	8.7E+7	100	769	8.2E+6	7	41 666	7.5E+8	100	806	1.6E+7
	ISG	784	100	1 113	8.9E+7	100	86	6.3E+5	100	132	2.3E+6	100	55	4.2E+6
HIFF	Dec. 2 (l.10)	2048	100	1 131	4.2E+7	100	3 395	6.0E+5	100	2 529	4.9E+6	100	225	5.5E+6
	Bim.10 (l.3)	1000	100	329	2.0E+7	0	N/A	N/A	0	N/A	N/A	100	10 067	5.6E+8
	St.Dec. 3 (l.4)	2401	100	2 473	7.5E+7	20	35 074	2.6E+7	100	17 113	1.3E+8	100	1 854	4.4E+7

how this mutation has influenced the FIHC result. All genes with different values are considered to be dependent on one another, and the gene  $x_m$  even if the original value of  $x_m$  is restored after FIHC. Each time the *linkage scrap* gathering procedure is triggered, the optimization order is randomly chosen (the same for  $\vec{x}$  and  $\vec{x}^{(m)}$ ). The requirement that the optimization order is the same for  $\vec{x}$  and  $\vec{x}^{(m)}$  is crucial. Otherwise, 3LO may produce a *false linkage*.

3LO never reports *false linkage*, but it may still miss some of the gene dependencies. To overcome this issue, *linkage scraps* are produced from many different genotypes and are used to construct the linkage trees in the way similar to the Linkage Tree Genetic Algorithm (LTGA) [6] and Parameter-less Population Pyramid (P3) [1]. First, the gene dependencies matrix, equivalent to the Dependency Structure Matrix (DSM) [1, 6], is constructed. For each gene pair, the number of occurrences in a single *linkage scrap* is counted. This simple measure replaces statistical-based measures. The main difference is that LTGA and P3 always build a single tree, which incorporates all of the genes. 3LO may produce many trees if there are no dependencies found between the genes that were assigned to two different trees.

The basic 3LO procedure described above is sufficient for solving problems built from fully-separable blocks or to gather linkage for overlapping problems (eg., Ising Spin Glass and NK landscapes). To properly discover linkage when solving problems with many levels of dependency, 3LO checks inter-block dependencies. Since the number of available values for a block of genes is large, they are filtered using the best-found individual. The details may be found in [5].

Based on the research an detailed analysis presented in [5], to effectively solve overlapping problems, it is not enough to have a precise linkage. A method must also use a diverse linkage. To achieve this requirement, the method employing 3LO, namely the 3LO Algorithm (3LOa), employs many populations, each of them maintaining separate linkage information (in the form of a linkage tree). To the best of our knowledge, 3LOa is the first method that

employs the multi-population scheme to increase linkage diversity, instead of increasing the population diversity.

## 2 THE RESULTS AND FUTURE WORK

3LOa was tested against a wide range of problems, including deceptive functions, overlapping problems, and problems with a high level of hierarchy. Its performance was compared with LT-GOMEA, P3, and the population-sizing DSMGA-II (psDSMGA-II) [2]. 3LOa was the only method that has found an optimal result for every considered test case in all runs. Partial results for some of the considered test cases are presented in Table 1. Complete results, including the scalability analysis, may be found in [5].

Although the concept of 3LO is in its initial stage, it may lead to excellent results. 3LOa is competitive with other state-of-the-art methods. Due to the fundamental differences between 3LO and other linkage learning techniques, 3LO will never support a *false linkage*. The main future work objectives are to limit the cost of 3LO and make it usable for other evolutionary methods (eg., LTGA, DSMGA-II, and P3) and hybridize 3LO with DSM-using methods.

## ACKNOWLEDGMENTS

This work was supported by the Polish National Science Centre (NCN) under Grant 2015/19/D/ST6/03115.

## REFERENCES

- [1] Brian W. Goldman and William F. Punch. 2014. Parameter-less Population Pyramid. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation (GECCO '14)*. ACM, 785–792.
- [2] Marcin M. Komarnicki and Michal W. Przewozniczek. 2019. Parameter-Less, Population-Sizing DSMGA-II. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO'19)*, 289–290.
- [3] Halina Kwasnicka and Michal Przewozniczek. 2011. Multi Population Pattern Searching Algorithm: A New Evolutionary Method Based on the Idea of Messy Genetic Algorithm. *IEEE Trans. Evolutionary Computation* 15 (2011), 715–734.
- [4] Martin Pelikan, Kumara Sastry, Martin V. Butz, and David E. Goldberg. 2006. Hierarchical BOA on Random Decomposable Problems. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO'06)*, 431–432.
- [5] Michal W. Przewozniczek and Marcin M. Komarnicki. 2020. Empirical Linkage Learning. *IEEE Transactions on Evolutionary Computation* (2020), (in press).
- [6] Dirk Thierens and Peter A.N. Bosman. 2013. Hierarchical Problem Solving with the Linkage Tree Genetic Algorithm. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation (GECCO'13)*, 877–884.