Exploring the Application of GOMEA to Bit-string GE

Eric Medvet
DIA - Università di Trieste
Trieste, Italy
emedvet@units.it

Alberto Bartoli
DIA - Università di Trieste
Trieste, Italy
bartoli.alberto@units.it

Andrea De Lorenzo
DIA - Università di Trieste
Trieste, Italy
andrea.delorenzo@units.it

ABSTRACT

We explore the application of GOMEA, a recent method for discovering and exploiting the model for a problem in the form of linkage, to Grammatical Evolution (GE). GE employs an indirect representation based on familiar bit-string genotypes and is applicable to any problem where the solutions may be described using a context-free grammar, which hence greatly favors its wide adoption. Being general purpose, the representation of GE raises the opportunity for benefiting from the potential of GOMEA to automatically discover and exploit the linkage. We analyze experimentally the application of GOMEA to two bit-string-based variants of GE representation (the original representation and the recent WHGE) and show that GOMEA is clearly beneficial when coupled to WHGE, whereas it delivers no significant advantages when coupled with GE.

CCS CONCEPTS

• **Theory of computation** → **Genetic programming**; *Grammars and context-free languages*; • **Computing methodologies** → **Heuristic function construction**; • **Mathematics of computing** → *Evolutionary algorithms*;

KEYWORDS

Genetic Programming, Linkage, Family of Subsets, Representation

ACM Reference Format:
Eric Medvet, Alberto Bartoli, and Andrea De Lorenzo. 2018. Exploring the Application of GOMEA to Bit-string GE. 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/10.1145/3205651.3205765

1 INTRODUCTION

Evolutionary Algorithms (EAs) are greatly appreciated also because they do not require their users to provide a model for the problem at hand: it is the EA itself which should discover, through the evolution, the model, i.e., how the parts of the solution interact. However, this result in not always achieved in practice. With the aim of addressing this limitation, in the last years a novel model-based EA, called Gene-pool Optimal Mixing Evolution Algorithm (GOMEA) [7], has been introduced. The contribution of GOMEA is twofold and consists of (1) a way, called Family of Subsets (FOS), to represent the model (called *linkage*) together with a method for learning that linkage from the population, and (2) a genetic operator called Gene-pool Optimal Mixing (GOM) in which the individual iteratively receives from random donors some portions of the genetic material defined by the FOS.

We here extend GOMEA to Grammatical Evolution (GE) [6], a popular and very widespread EA. GE well fits the design goals of GOMEA: in facts, with GE, a practitioner can tackle any problem whose solutions may be described by means of a context-free grammar (CFG) and is not required to know and tune the internals of the EA; as a consequence, the possible model underlying the problem is not available to GE.

Internally, GE is based on an indirect representation: genetic operators are applied to bit-string genotypes; then, bit-strings are transformed into solutions (i.e., strings of the language defined by the problem-specific CFG) by means of a genotype-phenotype mapping function. The indirect representation contributed to the success of GE, but has also been criticized for its poor properties [3, 8], whose improvement, in turn, motivated some recent variants for the individual representation, e.g., Weighted Hierarchical GE (WHGE) [4].

In this paper, we explore the application of GOMEA to two variants of GE—the original GE and WHGE—based on bit-string genotypes, but differing in the genotype-phenotype mapping: the latter is very relevant from the point of view of the linkage, with is discovered by GOMEA at the genotype level. We present the results of an experimental analysis involving GE, WHGE and four ways to model the linkage applied to four benchmark problems: we show that GOMEA does improve the search effectiveness when coupled with WHGE, whereas its combination with GE delivers mixed results.

Because of its demonstrated effectiveness on discrete optimization problems, GOMEA has been extended to other EAs or conditions, e.g., Genetic Programming in GOMEA-GP [9] and real-valued optimization in RV-GOMEA [1]. To the best of our knowledge, our proposal is the first application of GOMEA to GE. There have been, however, other studies aiming at exploiting, in GE, the knowledge of the model underlying the problem. Significant ones include [5] and [10], where the aim is to modify the grammar to discover new problem-specific building blocks and hence improve the search effectiveness. Somehow similarly, in [2] a method for obtaining a model (in the form of a graph) automatically from a grammar is presented.

2 GOMEA FOR BIT-STRING GE

Our proposal for the application of GOMEA to GE is based on GP-GOMEA [9]. After the initialization of the population, a main loop is repeated until a termination criterion is met and consists in two steps: (i) learning the linkage from the current population and

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

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3205765

(ii) applying the Gene-pool Optimal Mixing (GOM) variation operator to each individual in the population. The linkage is expressed as a Family of Subsets (FOS) $\mathcal{F} = \{F_1, F_2, \dots\}$ which is a set of sets of zero-based genotype indexes (*loci*): i.e., $F_i \subseteq \{0, \dots, l_g - 1\}$, where l_g is the evolution-wise immutable size of the genotype. We experimented with 4 different way of obtaining the FOS: 2 consisting in learning the FOS from the population—Linkage Tree (LT) and Random Tree (RT), both described in [9]—and 2 predefined FOS—Univariate (U) and Natural (N). Univariate assumes that there is no linkage between portions of the genotype: the FOS contains one singleton set for each possible locus, i.e., $\mathcal{F}_{\text{U}} = \{\{0\}, \{1\}, \dots, \{l_g - 1\}\}$. Natural captures the representation-dependent linkage: we applied it only to original GE, where the derivations are chosen using groups of 8 consecutive bits, i.e., $\mathcal{F}_{\text{N,GE}} = \{\{0, 1, \dots, 7\}, \{8, 9, \dots, 15\}, \dots\}$.

The application of the GOM operator to an individual (g, p, f) consists in repeating the following steps for each set F in a random permutation of $\mathcal{F}$: (i) a donor (g_d, p_d, f_d) is randomly chosen in the population and (ii) the portions of the genotype g defined by F are replaced with the corresponding portions coming from g_d; (iii) the fitness of the new individual is computed and, (iv) if there is a strict improvement, the modification on the individual g is kept, otherwise, it is rolled back.

We observed that, due to the high degeneracy, applying the GOM resulted often in no modifications to the individual. We hence included a *forced mutation* to be applied when the phenotype did not change after the processing of all the sets in $\mathcal{F}$. Moreover, in order to preserve good individual from being "forgot" due to the forced mutation, we also included a simple mechanism for keeping track of the best individual, updated at each iteration.

3 EXPERIMENTS AND CONCLUSIONS

We considered 4 benchmark problems—Parity-7, Nguyen-7, KLandscapes-5, and Text [3]—and used a prototype Java implementation (available at https://github.com/ericmedvet/evolved-ge) for both the standard EA (the *baseline*) and GOMEA on both GE and WHGE. We performed 30 runs for each combination of EA, FOS (only for GOMEA), representation, and problem, each run using a machine with 2 Intel Xeon E5-2694 v4 CPUs (2.3 GHz) with 18 cores each and 128 GB of RAM, and with the following parameters: genotype size $l_g = 512$ (GE) or $l_g = 128$ (WHGE); population size $n_{\text{pop}} = 500$; two-points same crossover with rate 0.8; bit-flip mutation with $p_{\text{mut}} = 0.01$ with rate 0.2; tournament selection of size 3; and max elapsed time $T_{\text{max}} = 60$ s as termination criterion.

Table 1 shows the mean and the standard deviation (across the 30 runs) of the final best fitness for each problem and variant: the table also shows, for each GOMGE variant and problem, the statistical significance (p-value with the Mann-Whitney U-test) of the null hypothesis that the final best fitness values have equal median of those obtained with the standard EA.

It can be seen that GOMEA, when coupled with WHGE, gives better results than the baseline in all the cases (FOS/problem) with the exception of U on KLandscapes-5: the difference is always statistically significant. In the Parity-7 problem, the increase in effectiveness is particularly clear: WHGE+RT and WHGE+LT obtain the perfect fitness in all the runs, whereas the baseline WHGE does not. On the other hand, the standard GE representation seems to be unsuitable for exploiting the advantages of GOMEA, in general: no significant difference are visible on one problem (Parity-7), a decrease in the fitness is visible on two problems (Nguyen7 and KLandscapes-5), and an improvement is visible for the last problem (Text).

Table 1: Mean and standard deviation of the final best fitness. Statistical significance (see text): ‡, †, and * stays for p lower than 0.01, 0.05, and 0.1, respectively.

	Var.	Parity-7	Nguyen7	KLand.-5	Text
GE	Base.	0.5 ±0.02	0.39±0.25	0.61±0.09	4.9±1.2
	U	0.5 ±0.01	0.49±0.19‡	0.63±0.06‡	3.5±0.7‡
	N	0.5 ±0	0.4 ±0.2	0.61±0.09	**3.1**±0.8‡
	RT	0.49±0.02	0.68±0.6 ‡	0.68±0.04‡	4.5±0.6‡
	LT	0.49±0.03	0.68±0.16‡	0.68±0.04‡	4.6±0.5‡
WHGE	Base.	0.17±0.13	0.52±0.19	0.4 ±0.08	5.7±0.8
	U	0.16±0.07*	0.31±0.15‡	0.6 ±0.05‡	4.9±0.5‡
	RT	**0** ±0 ‡	**0.18**±0.11‡	0.29±0.04‡	4 ±0 ‡
	LT	**0** ±0 ‡	0.21±0.12‡	**0.25**±0.07‡	4 ±0 ‡

In conclusion, we think that GOMEA may deliver significant performance improvements when coupled with WHGE, the most recent representation variant of GE. Despite further analysis has to be performed to confirm and explain the experimental findings here presented, we think that our results (1) suggest that the applicability of GE may be further improved by incorporating GOMEA and (2) shed new light on the possibility of WHGE to exhibit "good" and learnable linkage.

REFERENCES

[1] Anton Bouter, Tanja Alderliesten, Cees Witteveen, and Peter AN Bosman. 2017. Exploiting linkage information in real-valued optimization with the real-valued gene-pool optimal mixing evolutionary algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 705–712.

[2] Pei He, Zelin Deng, Chongzhi Gao, Liang Chang, and Achun Hu. 2017. Analyzing Grammatical Evolution and πGrammatical Evolution with Grammar Model. In *Information Technology and Intelligent Transportation Systems*. Springer, 483–489.

[3] Eric Medvet. 2017. A Comparative Analysis of Dynamic Locality and Redundancy in Grammatical Evolution. In *European Conference on Genetic Programming*. Springer, 326–342.

[4] Eric Medvet. 2017. Hierarchical Grammatical Evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '17)*. ACM, New York, NY, USA, 249–250.

[5] Michael O'Neill and Conor Ryan. 2004. Grammatical evolution by grammatical evolution: The evolution of grammar and genetic code. *Genetic Programming* (2004), 138–149.

[6] Conor Ryan, JJ Collins, and Michael O'Neill. 1998. Grammatical evolution: Evolving programs for an arbitrary language. In *European Conference on Genetic Programming*. Springer, 83–96.

[7] Dirk Thierens and Peter A.N. Bosman. 2011. Optimal Mixing Evolutionary Algorithms. In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation (GECCO '11)*. ACM, New York, NY, USA, 617–624.

[8] Ann Thorhauer. 2016. On the Non-uniform Redundancy in Grammatical Evolution. In *International Conference on Parallel Problem Solving from Nature*. Springer, 292–302.

[9] Marco Virgolin, Tanja Alderliesten, Cees Witteveen, and Peter A N Bosman. 2017. Scalable genetic programming by gene-pool optimal mixing and input-space entropy-based building-block learning. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 1041–1048.

[10] Pak-Kan Wong, Man-Leung Wong, and Kwong-Sak Leung. 2016. Hierarchical Knowledge in Self-Improving Grammar-Based Genetic Programming. In *International Conference on Parallel Problem Solving from Nature*. Springer, 270–280.