Dynamic Observation of Genotypic and Phenotypic Diversity for Different Symbolic Regression GP variants

Michael Affenzeller[1,2]
Stephan M. Winkler[1]
Bogdan Burlacu[1,2]
Gabriel Kronberger[1]
Michael Kommenda[1,2]
Stefan Wagner[1]

[1]University of Applied Sciences Upper Austria
Softwarepark 11, 4232 Hagenberg, Austria

[2]Johannes Kepler University Linz
Altenberger Straße 69, 4040 Linz, Austria

{maffenze,swinkler,bburlacu,gkronber,mkommend,swagner}@heuristiclab.com

ABSTRACT

Understanding the relationship between selection, genotype-phenotype map and loss of population diversity represents an important step towards more effective genetic programming (GP) algorithms. This paper describes an approach to capture dynamic changes in this relationship. We analyze the frequency distribution of points in the diversity plane defined by structural and semantic similarity measures. We test our methodology using standard GP (SGP) on a number of test problems, as well as Offspring Selection GP (OS-GP), an algorithmic flavor where selection is explicitly focused towards adaptive change. We end with a discussion about the implications of diversity maintenance for each of the tested algorithms. We conclude that diversity needs to be considered in the context of fitness improvement, and that more diversity is not necessarily beneficial in terms of solution quality.

CCS CONCEPTS

•Computing methodologies → Heuristic function construction; Randomized search;

KEYWORDS

Symbolic Regression, Genetic Programming, Population Dynamics, Genetic and Phenotypic Diversity, Offspring Selection

ACM Reference format:
Michael Affenzeller[1,2], Stephan M. Winkler[1], Bogdan Burlacu[1,2], Gabriel Kronberger[1], Michael Kommenda[1,2], and Stefan Wagner[1]. 2017. Dynamic Observation of Genotypic and Phenotypic Diversity for Different Symbolic Regression GP variants. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 6 pages.

GECCO '17 Companion, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3067695.3082530

DOI: http://dx.doi.org/10.1145/3067695.3082530

1 INTRODUCTION

Genetic Programming (GP) [3, 6] is a population-based evolutionary algorithm where solution candidates are gradually improved via the iterative application of selection and recombination.

Similar to biology, the ability to improve (ie., produce solutions of increased quality) inherently depends on the amount of genetic variation available in the population. A more diverse gene pool increases the chances of successful adaptation; thus, population diversity is essential to the algorithm's chances of success. However, maintaining population diversity in the presence of selection pressure remains an unresolved issue. For example, Xie [11] shows that loss of population diversity in GP is entirely due to the not-sampled individuals by selection.

Loss of diversity under fitness-based selection is considered to be the main cause of premature convergence, a situation where the offspring produced by the algorithm are no longer able to outperform their parents. Since selection acts on phenotypes, it becomes necessary to investigate diversity loss starting from underlying factors such as selection pressure, the antagonistic relationship between exploration and exploitation, and the non-injective (many-to-one) mapping from genotypes to phenotypes.

Our motivation, in this context, is to analyze population diversity at both the genotype and phenotype levels and investigate possible connections between the two. We consider tree-based GP where the genotypes are represented by symbolic expression trees. We employ structural (genotypic) and semantic (phenotypic) similarity measures and describe the dynamical evolution of diversity in the "similarity plane" formed by these two axes. We test different algorithmic flavors and selection mechanisms in order to analyze their effects on GP population dynamics.

The remainder of this paper is organized as follows: Section 2 introduces our similarity measures and methodology, Section 3 details the experiment configuration, Section 4 describes the used problems and Section 5 is dedicated to conclusions.

2 SIMILARITY MEASURES

We here introduce a new genotype similarity measure based on the bottom-up tree distance [7] and a phenotype similarity measure based on the correlation between two individuals' outputs.

Since our similarity measures are symmetrical, the number of similarity calculations necessary to compute the average similarity for a population of N individuals is $\frac{N(N-1)}{2}$. Therefore, the population diversity is given by:

$$Div(T) = 1 - \frac{\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} Sim(t_i, t_j)}{N(N-1)/2} \tag{1}$$

where $Sim(t_1, t_2)$ can be either the bottom-up or the phenotypic similarity.

2.1 Genotypic Similarity

Genotypic similarity is calculated using a bottom-up tree mapping based on the largest common forest between trees, as described by [7]. It has the advantage of maintaining the same time complexity, namely linear in the size of the two trees regardless of whether the trees are ordered or unordered. The algorithm works as follows:

(1) In the first step, it computes the compact directed acyclic graph representation G of the largest common forest $F = t_1 \cup t_2$ (consisting of the disjoint union between the two trees). The graph G is built during a bottom-up traversal of F (in the order of non-decreasing node height). Two nodes in F are mapped to the same vertex in G if they are at the same height and their children are mapped to the same sequence of vertices in G. The bottom-up traversal ensures that children are mapped before their parents, leading to $O(|t_1| + |t_2|)$ time for adding vertices in G corresponding to all nodes in F. This step returns a map $K : F \to G$ which is used to compute the bottom-up mapping.
(2) The second step iterates over the nodes of t_1 in level-order and builds a mapping $M : t_1 \to t_2$ using K to determine which nodes correspond to the same vertices in G. The level-order iteration guarantees that every largest unmapped subtree of t_1 will be mapped to an isomorphic subtree of t_2.

Finally, the bottom-up similarity between trees t_1 and t_2 is calculated as

$$\text{BottomUpSimilarity}(t_1, t_2) = \frac{2 \cdot |M(t_1, t_2)|}{|t_1| + |t_2|} \tag{2}$$

By taking two times the size of the bottom-up mapping between the two trees, we make sure that the similarity values will always fall inside the $[0, 1]$ interval.

2.2 Phenotypic Similarity

We define an individual's phenotype as its evaluation response on the training data. Individuals with the same response (with the same *semantics*) are considered phenotypically similar regardless of their actual structure. In this paper, we introduce a phenotypic similarity measure based on the squared Pearson product-moment correlation coefficient:

$$R^2_{X,Y} = \left(\rho_{X,Y}\right)^2 = \left(\frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}\right)^2 \tag{3}$$

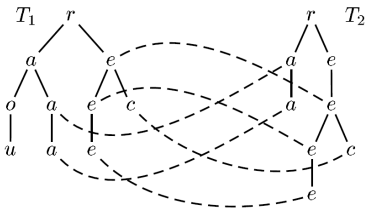


Figure 1: Bottom-up mapping between two trees t_1 and t_2 [7]

Since $\rho \in [-1, +1]$, the R^2 correlation coefficient will always return a similarity value in the interval $[0, 1]$.

To avoid undefined situations where the denominator is zero in the above formula, we introduce an exception for the case when one of the responses has variance zero. Two individuals with constant responses are considered to be completely similar; otherwise, if only one of them has a constant response, similarity is set to zero.

$$\text{PhenotypicSim}(t_1, t_2) = \begin{cases} 1 & \text{if } \text{Var}(t_1) = \text{Var}(t_2) = 0 \\ 0 & \text{if } \text{Var}(t_1) = 0 \text{ or } \text{Var}(t_2) = 0 \\ R^2_{t_1,t_2} & \text{otherwise} \end{cases} \tag{4}$$

3 EXPERIMENTAL SETUP

We analyze the effect of selection on population dynamics for the standard GP algorithm and Offspring Selection GP (OS-GP). We configure each algorithm using typical and competitive parameter settings and test with two problem instances (one synthetic and one real-world problem) which we describe in the next section.

3.1 Algorithms

3.1.1 Standard Genetic Programming (SGP). We configure the standard genetic programming algorithm with the following parameter settings:

- Population size: 500 individuals
- Termination criterion: 1000 generations
- Tree initialization: probabilistic Tree Creation (PTC2) ([4])
- Maximum tree size: 50 nodes, 10 levels
- Elites: 1 individual
- Parent selection: tournament selection, group size 5
- Crossover: subtree crossover, 100% probability
- Mutation: 25% mutation rate, each mutation is performed either as single-point, multi-point, remove branch or replace branch mutation
- Fitness function: coefficient of determination R^2 ([2])
- Terminal symbols: constant, weight * variable
- Function symbols: binary functions (+,-,×, ÷, exp, log)

3.1.2 Genetic Programming with Offspring Selection (OSGP). Strict offspring selection (OS) [1] shifts the focus of selection towards adaptive change by introducing an additional selection step where newly created individuals are accepted into the population only if their fitness exceeds that of their parents. The algorithm produces as many individuals as needed in order to fill in a new

generation of individuals. In this context, the *active selection pressure* is defined as the ratio between the total number of produced offspring and the number of individuals needed to fill a generation (ie., the population size). The active selection pressure varies every generation depending on how easy it is to generate better offspring. The active selection pressure at generation i is expressed as:

$$SelectionPressure(i) = \frac{|GeneratedOffspring(i)|}{|Population|}. \quad (5)$$

We use the selection pressure as termination criterion, ie., the algorithm is terminated as soon as the selection pressure reaches a predefined maximum value.

Most parameters for these OS-GP tests are equal to those used for standard GP; OS-GP specific parameter settings are changed as follows:

- Population size: 200 individuals
- Termination criterion: Maximum selection pressure 200
- Parent selection: Gender specific ([9]); proportional and random
- Offspring selection: Strict, i.e. success ratio = 1.0 and comparison factor = 1.0 ([1])

3.2 Problem Instances

We test the aforementioned GP algorithms on two benchmark regression problems to examine population dynamics. The problems are taken from the recommended GP benchmark problems [10] and are both available within the HeuristicLab framework.

- The *Poly-10* data set [5] consists of 500 samples with 10 variables $x_{1\ldots10}$ and the response variable y. The values $x_{1\ldots10}$ were generated by randomly (uniformly) drawing values from the interval $[-1, +1]$, the response values were calculated according to the following equation:

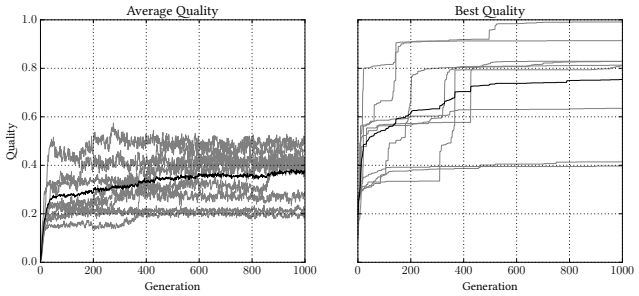
$$y = f(\mathbf{x}) = x_1x_2 + x_3x_4 + x_5x_6 + x_1x_7x_9 + x_3x_6x_{10}$$

- The Tower data set [8] comes from an industrial problem on modeling gas chromatography measurements of the composition of a distillation tower. It contains 5000 records and 25 potential input variables, the target variable is the propylene concentration at the top of the distillation tower. The samples were measured by a gas chromatograph and recorded as floating averages every 15 minutes. The 25 potential inputs are temperatures, flows, and pressures related to the distillation tower. The Tower data set can be downloaded from http://www.symbolicregression.com/?q=towerProblem.

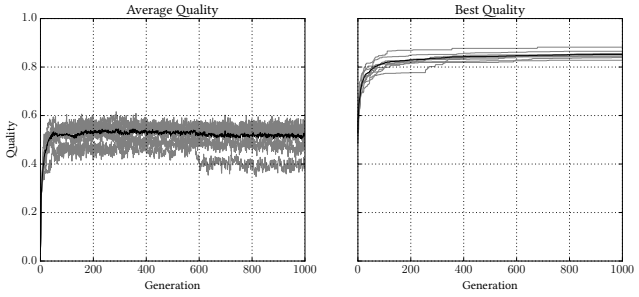
4 EXPERIMENTAL RESULTS

We build our similarity histograms using measurements acquired from 10 runs for each problem instance (*Poly-10* and *Tower*) and algorithmic configuration. We create a visual illustration of the evolution of similarity with bidimensional similarity histograms smoothed using gaussian kernel density estimation. Additionally, we plot the average population fitness for each algorithm and problem combination, in order to investigate whether the evolution of population similarity is related to the evolution of average fitness.

Figures 2 and 3 show the evolution of average population quality, while Figures 4 and 5 show the evolution of similarities on

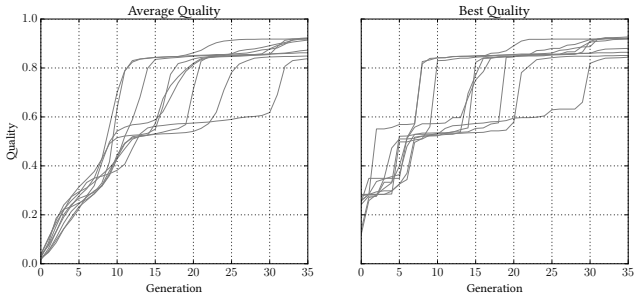


(a) *Poly-10* Problem

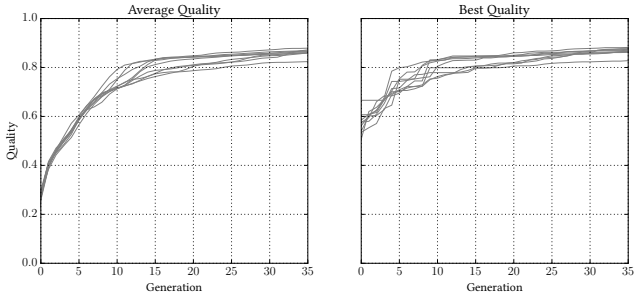


(b) *Tower* Problem

Figure 2: Standard GP average and best population quality



(a) *Poly-10* Problem



(b) *Tower* Problem

Figure 3: OS-GP average and best population quality

the two benchmark problems for the GP and OS-GP algorithms, respectively. The subfigures represent snapshots taken every 100 generations in the case of standard GP and every 5 generations in

the case of OS-GP. A comparison between the two figures leads us to the following observations:

- Phenotypic similarity is more heavily dependant on selection pressure and, indirectly, on the average fitness of the population. Strict offspring selection (where genetic changes can only improve fitness, otherwise the offspring gets rejected) causes all individuals in the population to become semantically similar.
- Semantically similar individuals tend to be structurally similar as well. This is noticeable on all figures by looking at the y-axis values towards the right-hand part of each histogram.
- Semantic similarity increases with quality. This is particularly noticeable for the tested standard GP instance (Figures 2b and 2a) where a higher average population quality (*Tower* problem) translates into higher phenotypic similarity (Figures 4b and 4a).

The significant difference in the evolution of similarities between GP and OS-GP suggests that diversity loss is more pronounced when selection is focused on adaptive changes. Therefore, strict offspring selection determines the evolution of a more homogeneous population at both structural and semantic levels.

We notice that maintaining diversity is useful only in situations where the algorithm is able to exploit it to obtain better solution candidates. In practice, this does not always seem to be the case. On one hand, the obtained qualities show that the considered standard GP instance is unable to exploit the additional diversity in the population. This seems to be caused by a wider variation of fitness in the population, where low-fitness individuals contribute diversity but their genetic material is not necessarily useful.

On the other hand, strict offspring selection in OS-GP leading to increased genotypic and phenotypic similarity allows the algorithm to make better use of the available genetic variation, translating into higher average population fitness and better overall results.

5 CONCLUSION

The goal of this work is to show the potential of dynamic genotype/phenotype similarity observations exemplarily on the basis of two algorithm instances and two benchmark problems. Our analysis reveals the important role played by selection in genetic programming.

In future work we plan to extend this approach on other problems and algorithmic configurations, with different parent and child selection mechanisms. Additionally, we plan to develop improved visualization methods which show more clearly the relationship between the evolution of diversity and the evolution of population quality.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support by the Austrian Research Promotion Agency (FFG) within the COMET Project Heuristic Optimization in Production and Logistics (HOPL), #843532.

REFERENCES

[1] Michael Affenzeller, Stephan Winkler, Stefan Wagner, and Andreas Beham. 2009. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications.* CRC Press, Singapore. http://gagp2009.heuristiclab.com/

[2] Norman R. Draper and Harry Smith. 1998. *Applied Regression Analysis, 3^{rd} edition.* Wiley-Interscience.

[3] John R. Koza. 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA.

[4] Sean Luke. 2000. Two Fast Tree-Creation Algorithms for Genetic Programming. *IEEE Transactions on Evolutionary Computation* 4, 3 (Sept. 2000), 274–283.

[5] Riccardo Poli. 2003. A Simple but Theoretically-motivated Method to Control Bloat in Genetic Programming. In *Genetic Programming, Proceedings of EuroGP'2003 (LNCS)*, Conor Ryan, Terence Soule, Maarten Keijzer, Edward Tsang, Riccardo Poli, and Ernesto Costa (Eds.), Vol. 2610. Springer-Verlag, Essex, 204–217.

[6] Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. 2008. *A field guide to genetic programming.* Published via `http://lulu.com` and freely available at `http://www.gp-field-guide.org.uk`.

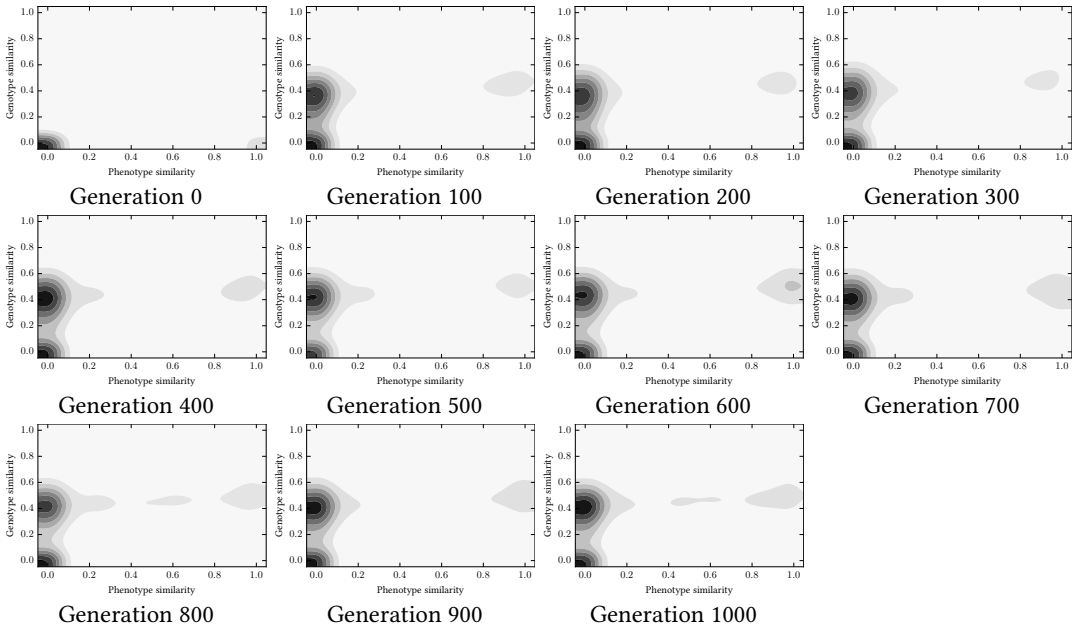
[7] Gabriel Valiente. 2001. An efficient bottom-up distance between trees. In *Proceedings of the 8th International Symposium of String Processing and Information Retrieval.* Press, 212–219.

[8] Ekaterina J Vladislavleva, Guido F Smits, and Dick Den Hertog. 2009. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *Evolutionary Computation, IEEE Transactions on* 13, 2 (2009), 333–349.

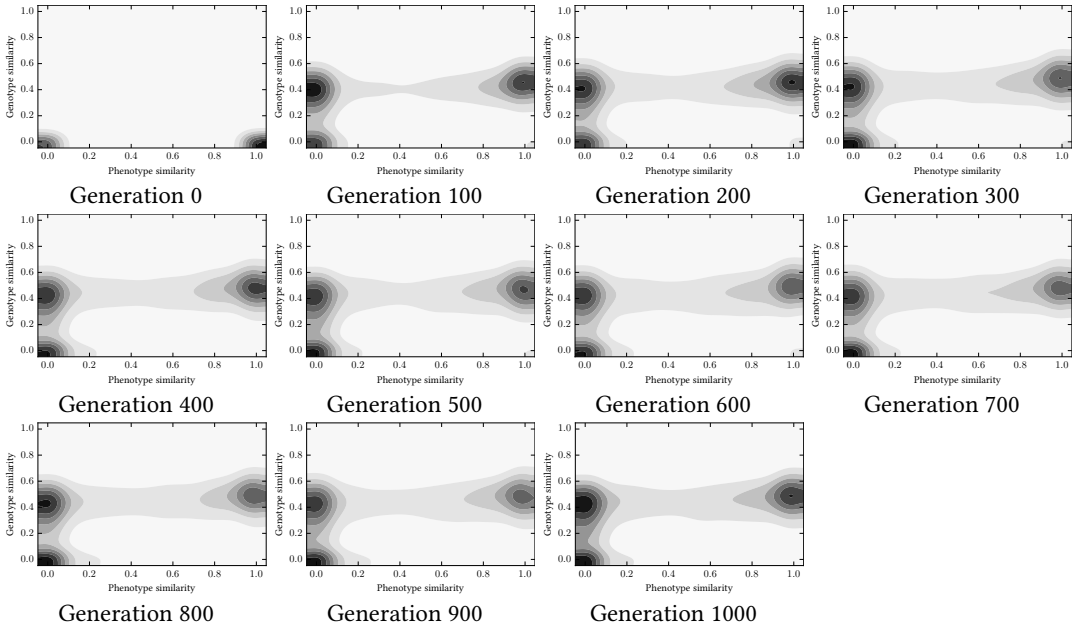
[9] S. Wagner and M. Affenzeller. 2005. SexualGA: Gender-Specific Selection for Genetic Algorithms. In *Proceedings of the 9^{th} World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI) 2005*, N. Callaos, W. Lesso, and E. Hansen (Eds.), Vol. 4. International Institute of Informatics and Systemics, 76–81.

[10] David R. White, James McDermott, Mauro Castelli, Luca Manzoni, Brian W. Goldman, Gabriel Kronberger, Wojciech Jaskowski, Una-May O'Reilly, and Sean Luke. 2013. Better GP benchmarks: community survey results and proposals. *Genetic Programming and Evolvable Machines* 14, 1 (March 2013), 3–29. `DOI:` http://dx.doi.org/doi:10.1007/s10710-012-9177-2

[11] Huayang Xie. 2008. *An Analysis of Selection in Genetic Programming.* Ph.D. Dissertation. Computer Science, Victoria University of Wellington, New Zealand. http://homepages.ecs.vuw.ac.nz/~mengjie/students/jasonPhd_thesis.pdf

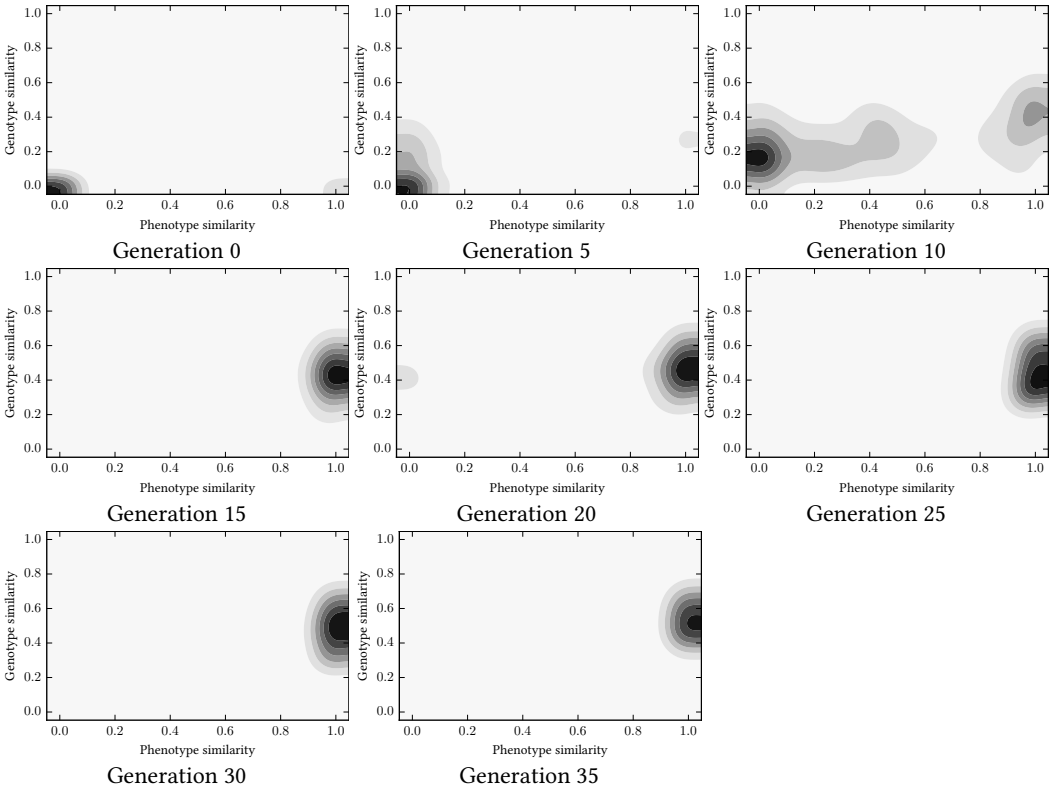


(a) *Poly-10* Problem

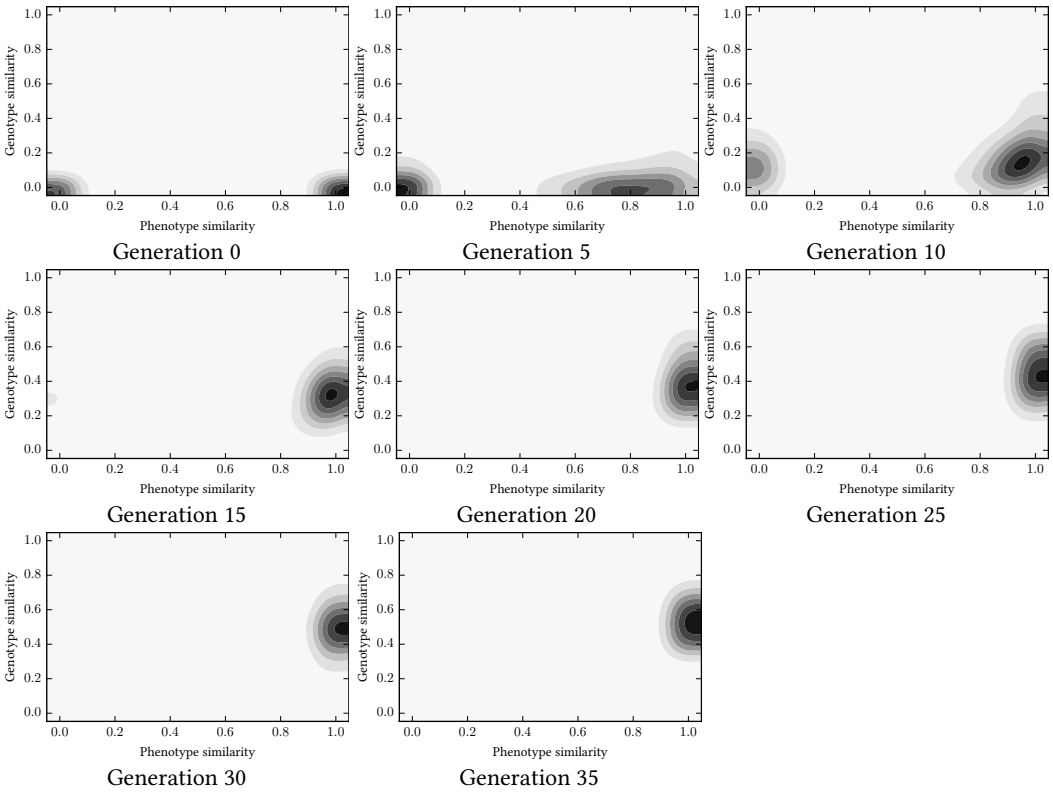


(b) *Tower* Problem

Figure 4: Distribution of genotypic vs. phenotypic similarities in standard GP



(a) *Poly-10* Problem



(b) *Tower* Problem

Figure 5: Genotypic vs. phenotypic similarities in OS-GP