Genetic and Evolutionary Computation

Rick Riolo · Bill Worzel · Brian Goldman
Bill Tozier *Editors*

Genetic Programming Theory and Practice XIV

Springer

Genetic and Evolutionary Computation

More information about this series at http://www.springer.com/series/7373

Rick Riolo • Bill Worzel • Brian Goldman
Bill Tozier
Editors

Genetic Programming Theory and Practice XIV

Springer

Editors
Rick Riolo
Center for the Study of Complex Sys
University of Michigan
Ann Arbor, MI, USA

Brian Goldman
Colorado State University
Fort Collins, CO, USA

Bill Worzel
Evolution Enterprises
Ann Arbor, MI, USA

Bill Tozier
Ann Arbor
MI, USA

ISSN 1932-0167
Genetic and Evolutionary Computation
ISBN 978-3-319-97087-5 ISBN 978-3-319-97088-2 (eBook)
https://doi.org/10.1007/978-3-319-97088-2

Library of Congress Control Number: 2018957658

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This is the written record of the 14th meeting of the annual Genetic Programming Theory and Practice Workshop, which was hosted by the Center for the Study of Complex Systems at the University of Michigan, in Ann Arbor, May 19–21, 2016.

It is, as a matter of course, woefully incomplete. It can serve only as a fragmentary record of our meeting. I say this not as an apology, but rather as a sort of gentle warning, of a sort I rarely seem to see in these collections.

Let me try to explain. The central and explicit focus of the GPTP Workshop has always been the conversations that are fostered *at* the meeting itself. These conversations happen among the invited keynote speakers, the technical participants, the students, and sponsors who spend somewhat more than 3 days in a room together. The technical work that invited speakers have prepared *before* they arrive, and which they bring with them to present in session and discuss, is really just the provocation or *seed* of the "real workshop." As a result, the proceedings volume from each workshop should only ever be read as a record of where some subset of attendees started individually, not as a position in which we ended up as a group by the end. The *real* results will only gradually appear in the subsequent literature, as a cloud of works that may not even explicitly refer to this meeting, and in the subtly changed directions of ongoing research programs in years to come.

In other words, think of these chapters as the *inputs* to a years-long dynamical process, not as the final *output* of that process.

Further, this volume does not even manage to cover all of the presentation-driven portion of our discussions. It is a sad but unavoidable truth of modern worklife that only a fraction of our most influential participants have the spare time to produce a chapter for you to read. Our three invited keynote speakers, for example, inevitably have a strong influence on our collective attention, setting the tone and focus of the entire week as soon as they begin speaking... but they are rarely able to make time to provide written contributions here. And because we so actively seek out both industrial and academic speakers, we often hear at least one or two presentations that treat contemporary professional work from an industrial domain. Proprietary and ongoing projects cannot always be translated successfully into an academic-style paper.

Since this was our 14th meeting, you should suspect that even calling these chapters the "inputs" of a long-term process misses the preceding history. If we can only detect the "outputs" in the workshop's future-facing light-cone—by seeing the consequences of the week's conversations and insights in the attendees' actions—then surely almost all of 2016s workshop was in turn influenced by earlier sessions. You will find chapters here describing the ongoing efforts of Michael Affenzeller's HEAL group, as they implement, explore, and distribute ideas and algorithms in their HeuristicLab system, which have in several cases arisen directly from earlier GPTP workshops. Michael Korns carries forward his many years' effort in support of industrial-strength symbolic regression systems. There are at least three chapters here from Lee Spector, Nic McPhee, Thomas Helmuth, and colleagues, which have arisen from many years of fruitful exploration of the Push language for GP, and so on. I doubt a single chapter lacks a strong link from some earlier GPTP conversation.

Therefore, think of this book as a collection of blurred snapshots, taken from the center of the web of conversations that make up the "real" GPTP Workshop. The historian interested in the context from which they have arisen should review the prior 13 volumes in the series *at least* as diligently as she reviews the more mundane published conference literature in adjacent years.

Assume that we took the time to speak to one another—substantively, but not *constantly*—about the material presented in this volume. But in the moment and in the room together, our conversations were much more along the lines of, "That's fascinating, but have you considered…," or "I wonder if you've noticed that what you did in Section 3 could be related to what Smith did in her work on…," and so forth. We do not just clap and grumble at one another—we discuss. When it works, this workshop is a *generative process*, not a ritualized presentation. Most of the "work" it does in the field will have happened after the attendees arrived, in the lunch breaks, the hallway conversations, and the notes we have scribbled on a napkin in the pub afterward, or in our notebooks in hotel rooms after quiet thoughtful dinners.

In the following 14 chapters, you will find the subset of contributions from invited speakers who were able to provide them. They have done a good job contextualizing and framing their work, subject to the caveats I have spelled out above. So rather than simply revisiting each one in turn, let me try to fill in the gaps between them a bit, beginning with our three keynote speakers.

Dr. Joanna Masel, from Ecology and Evolutionary Biology at the University of Arizona, was the first of our invited keynote speakers. She spoke on "Evolution of molecular error rates, and the consequences for robustness, evolvability, and the de novo emergence of new protein-coding genes from junk DNA." She pointedly reminded the audience—we are for the most part computer science folks—of the deep fundamental differences between biological and "computational" evolutionary processes. In the course of her presentation, she did an excellent job conveying both the fascinating complexity of biological evolution (and evolutionary biology, the discipline), and also the potential shortcomings of our greatly simplified mental models in evolutionary computing. Throughout the workshop, she was able to

helpfully remind many speakers and participants of the ways in which such an overly glib simple model might lead one into a proverbial ditch.

Dr. Stephanie Forrest, from the University of New Mexico and the Santa Fe Institute (a colleague of several attendees), spoke in her keynote lecture about "Software: Evolution, Robustness, and Diversity (also, the Mutation Cliff)." She spoke about her group's and colleagues' ongoing research into the complexities of *real* (which is to say: human-written) software systems and in particular their work in the new field of genetic improvement and automated software repair. As with Dr. Masel's effort, we in the audience were frequently and pointedly reminded of the "simplifying assumptions" our work often makes for the sake of being tractable. Once again, there are many ways in which the consequences from our overly simplified framing notions can stumble, when faced with the externalities of the real world. In particular, she spoke of her own practical and philosophical explorations of what "fitness" might mean in the context of producing well-repaired software originally produced by human programmers. As my notes have it, "What is a *reasonable* way of quantifying the performance of a broken or repaired piece of low-level software infrastructure?" We tend in the field of GP not to think very often about bug reports, side effects, and other matters that live past the interface with software development, deployment, and usability... but our more advanced work inevitably bumps up against it.

On the third day of our meeting, Dr. Cosma Shalizi from the Statistics Department at Carnegie Mellon University and the Santa Fe Institute spoke on "Bayesian Learning, Evolutionary Search, and Information Theory." And boy did he. He pointed out remarkable (but as far as I am aware, previously unremarked) similarities in the deep structure of Bayesian learning representations and algorithms, the replicator equation and other core dynamical models from theoretical population biology, and the ways in which information (in Shannon's sense) is "handled" by these processes. In other words, in a bit more than an hour, he stitched together three increasingly independent disciplinary approaches to learning and dynamical systems models and described a strong framework for exploring what it might mean for evolution to "learn." This delightful presentation—which several of us hope he will have published somewhere soon—brought in frameworks from disciplines that will certainly benefit from an ontological reconciliation like this. In other words, we look forward to reading his paper on the matter *at least as much* as you do.

Intermingled with the three keynote speakers' talks, and the presentations from the invited speakers whose chapters follow, there were also innumerable breakfast, lunch, and dinner meetings (all ad hoc of course). Watch for their effects in the *future* work of the attendees, as you read the works they brought to the table to begin our conversations with one another.

Ann Arbor, MI, USA — William Tozier
Fort Collins, CO, USA — Brian Goldman
Ann Arbor, MI, USA — Rick Riolo
Ann Arbor, MI, USA — William P. Worzel
December 2017

Acknowledgments

We feel obliged not only to *thank* the participants in the 2016 GPTP Workshop but to emphasize how proud we are to have played our little role in bringing them together. We realize (because they say it over and over, year after year) that the guests think this is an enjoyable and inspiring meeting, but the pleasure is literally ours—at least in hindsight. If you listen, you will notice our anecdotes all talk about the way we watched you talk with one another so productively and pleasantly, not just what you said but how you *were* together. So thank you all for being interesting people who can now and then listen to one another.

The keynote speakers are not just punctuation marks or particularly honored members of our little community, but have been tasked with a crucial role of jostling our attention in a productive way and steering us away from our discipline-killing tendency for *only ever* talking to one another. Our speakers in 2016 did an admirable job in this, and we appreciate their company at least as much as the insights they brought from adjacent disciplines.

We could not undertake these ambitious workshops without the generous sponsorships, with whose help we are increasingly able to bring new colleagues into the fold. In 2016, these included:

- The Center for the Study of Complex Systems at the University of Michigan and especially Carl Simon and Charles Doering, the champions of the workshop series
- John Koza
- Jason Moore
- Babak Hodjat at Sentient
- Michael Korns and Gilda Cabral
- Mark Kotanchek at Evolved Analytics
- The Heuristic and Evolutionary Algorithms Laboratory at the Upper Austria University of Applied Science

Finally, there are those without whose help we would not have been able to manage: Linda Wood and Mita Gibson at the Center for the Study of Complex Systems.

Thank you all for making this possible, and a pleasure.

Contents

Contributors

Michael Affenzeller Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria

Itay Azaria Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Wolfgang Banzhaf Department of Computer Science, Memorial University, St. John's, NL, Canada

BEACON Center of the Study of Evolution in Action, Michigan State University, East Lansing, MI, USA

Armand R. Burks BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI, USA

Bogdan Burlacu Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria

Maggie M. Casale University of Minnesota, Morris, Morris, MN, USA

Mauro Castelli NOVA IMS, Universidade Nova de Lisboa, Lisbon, Portugal

Achiya Elyasaf Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Mitchell D. Finzel University of Minnesota, Morris, Morris, MN, USA

Philipp Fleck Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Steven Gustafson Maana, Bellevue, WA, USA

Thomas Helmuth Computer Science, Washington and Lee University, Lexington, VA, USA

Erik Hemberg MIT CSAIL, Cambridge, MA, USA

Babak Hodjat Sentient Technologies, San Francisco, CA, USA

Chris Holmes University of Oxford, Oxford, UK

Ting Hu Department of Computer Science, Memorial University, St. John's, NL, Canada

Perla S. Juárez-Smith Tecnológico Nacional de México/I.T. Tijuana, Tijuana, B.C., Mexico

Michael Kommenda Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria

Michael F. Korns Analytic Research Foundation, Henderson, NV, USA

Krzysztof Krawiec Institute of Computing Science, Poznan University of Technology, Poznań, Poland

Gabriel Kronberger Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Pierrick Legrand Universitè de Bordeaux, Institut de Mathèmatiques de Bordeaux, UMR CNRS 5251, Bordeaux, France

CQFD Team, Inria Bordeaux Sud-Ouest, Talence, France

Nicholas Freitag McPhee Division of Science and Mathematics, University of Minnesota, Morris, Morris, MN, USA

Risto Miikkulainen Sentient Technologies, San Francisco, CA, USA

Jason H. Moore Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA, USA

Luis Muñoz Tecnológico Nacional de México/I.T. Tijuana, Tijuana, B.C., Mexico

Lawrence Murray University of Oxford, Oxford, UK

Luiz Otavio V. B. Oliveira DCC, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

Randal S. Olson Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA, USA

Una-May O'Reilly MIT CSAIL, Cambridge, MA, USA

Fernando E. B. Otero School of Computing, University of Kent, Chatham Maritime, UK

Gisele L. Pappa DCC, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil

William F. Punch BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI, USA

Jacob Rosen MIT CSAIL, Cambridge, MA, USA

Oliver Schütze Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico

Hormoz Shahrzad Sentient Technologies, San Francisco, CA, USA

Saul Shanabrook Computer Science, University of Massachusetts, Amherst, MA, USA

Sara Silva BioISI Biosystems and Integrative Sciences Institute, Faculty of Sciences, University of Lisbon, Lisbon, Portugal

Moshe Sipper Department of Computer Science, Ben-Gurion University, Beer-Sheva, Israel

Lee Spector Cognitive Science, Hampshire College, Amherst, MA, USA

Arun Subramaniyan BHGE - Digital, San Ramon, CA, USA

Jerry Swan Department of Computer Science, University of York, York, UK

Leonardo Trujillo Tecnológico Nacional de México/I.T. Tijuana, Tijuana, B.C., Mexico

Leonardo Vanneschi NOVA IMS, Universidade Nova de Lisboa, Lisbon, Portugal

Stephan M. Winkler Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria

Aisha Yousuf Eaton Corporation, Southfield, MI, USA

Emigdio Z-Flores Tecnológico Nacional de México/I.T. Tijuana, Tijuana, B.C., Mexico

Chapter 1
Similarity-Based Analysis of Population Dynamics in Genetic Programming Performing Symbolic Regression

Stephan M. Winkler, Michael Affenzeller, Bogdan Burlacu, Gabriel Kronberger, Michael Kommenda, and Philipp Fleck

Abstract Population diversity plays an important role in the evolutionary dynamics of genetic programming (GP). In this paper we use structural and semantic similarity measures to investigate the evolution of diversity in three GP algorithmic flavors: standard GP, offspring selection GP (OS-GP), and age-layered population structure GP (ALPS-GP). Empirical measurements on two symbolic regression benchmark problems reveal important differences between the dynamics of the tested configurations. In standard GP, after an initial decrease, population diversity remains almost constant until the end of the run. The higher variance of the phenotypic similarity values suggests that small changes on individual genotypes have significant effects on their corresponding phenotypes. By contrast, strict offspring selection within the OS-GP algorithm causes a significantly more pronounced diversity loss at both genotypic and, in particular, phenotypic levels. The pressure for adaptive change increases phenotypic robustness in the face of genotypic perturbations, leading to less genotypic variability on the one hand, and very low phenotypic diversity on the other hand. Finally, the evolution of similarities in ALPS-GP follows a periodic pattern marked by the time interval when the bottom layer is reinitialized with new individuals. This pattern is easily noticed in the lower layers characterized by shorter migration intervals, and becomes less and less noticeable on the upper layers.

S. M. Winkler (✉) · M. Affenzeller · B. Burlacu · M. Kommenda
Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

Institute for Formal Models and Verification, Johannes Kepler University, Linz, Austria
e-mail: Stephan.Winkler@fh-hagenberg.at

G. Kronberger · P. Fleck
Heuristic and Evolutionary Algorithms Laboratory, University of Applied Sciences Upper Austria, Hagenberg, Austria

R. Riolo et al. (eds.), *Genetic Programming Theory and Practice XIV*, Genetic and Evolutionary Computation,
https://doi.org/10.1007/978-3-319-97088-2_1

Keywords Genetic programming · Symbolic regression · Population dynamics · Genetic diversity · Phenotypic diversity · Offspring selection · Age-layered population structure · ALPS

1.1 Introduction: Genetic Programming, Population Diversity, and Population Dynamics

Genetic Programming (GP) [6, 9] is a powerful optimization technique which evolves a population of tree-encoded solution candidates according to the rules of natural selection. Similar to its biological counterpart, the GP algorithm is dependent on the two steps of Darwinian evolution: variation (due to crossover and mutation) and selection. Genotypes (G) are mapped into phenotypes (P), an evaluation function $f : S \rightarrow \mathbb{R}$ (where S is the solution space) assigns a fitness value to each individual in the population. If a variation in a trait is more successful (i.e., it improves an organism's propagation success rate by allowing it to have more viable offspring) then that trait may eventually come to dominate the population. When certain phenotypic traits dominate the population to the detriment of others, genotypic variation with a high adaptive potential but lower fitness runs the risk of becoming extinct as a consequence of selection. Schaper and Louis [10] suggest this happens when the more "globally fit" do not have time to be found or to fix in the population over evolutionary timescales. The authors suggest that "strong biases in the rates at which traits can arrive through variation may direct evolution towards outcomes that are not simply the fittest". Thus, loss of diversity has a negative impact on the search by reducing the population's adaption potential.

While low genotypic diversity will in most cases hinder the genetic process to generate novel solution candidates, low phenotypic diversity might indicate that there is no significant search progress since newly created individuals are not better than their parents. This is why we here specifically analyze phenotypic as well as genotypic diversities, as this shall also enable a more detailed discussion about the reasons of premature convergence (seen in genotypic diversity) and its consequences (seen in phenotypic diversity).

In this paper, we analyze empirically the loss of population diversity for different GP flavors and symbolic regression problem instances. We introduce computational methods for measuring diversity at the genotypic and phenotypic level, and investigate the correlation between the two. Burke et al. [2] provide a good overview of various distance measures, analyzing the correlation between fitness and diversity; structural versus evaluation based solutions similarity analysis for symbolic regression was for example discussed in [16]. We show the progress of genotype and phenotype population diversity for three GP algorithmic configurations, namely standard GP, GP with strict offspring selection, and ALPS-GP.

The chapter is organized as follows: Sect. 1.2 describes the tree distance metrics that were used for similarity calculation and the methodology for our experiments. Section 1.3 describes the test settings, Sect. 1.4 summarizes the obtained results, and in Sect. 1.5 we give our conclusions.

1.2 Similarity Measures

We here introduce a new genotype similarity measure based on the bottom-up tree distance [11] 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:

$$Div(T) = 1 - \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} Sim(t_i, t_j)}{N(N-1)/2}, \tag{1.1}$$

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

1.2.1 Genotypic Similarity

Genotypic similarity is calculated using a measure similar to the tree edit distance, called the *bottom-up distance*. The bottom-up tree distance is a flexible distance measure based on the largest common forest between trees, as described by Valiente [11]. 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 distance between trees t_1 and t_2 is calculated as

$$\text{BottomUpDistance}(t_1, t_2) = \frac{2 \times |M(t_1, t_2)|}{|t_1| + |t_2|}. \tag{1.2}$$

Thus, the similarity of t_1 and t_2 is defined as

$$\text{GenotypicSimilarity}(t_1, t_2) = 1 - \text{BottomUpDistance}(t_1, t_2). \tag{1.3}$$

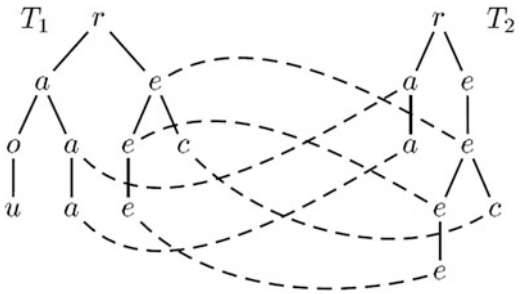

Fig. 1.1 Bottom-up mapping between two trees t_1 and t_2 (see [11])

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 (Fig. 1.1).

1.2.2 Phenotypic Similarity

Similarity at the phenotype level is calculated with regard to an individual's 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:

$$R^2_{X,Y} = \left(\rho_{X,Y}\right)^2 = \left(\frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y}\right)^2. \tag{1.4}$$

Since $\rho \in [-1, +1]$, the R^2 correlation coefficient will always return a similarity value in the interval [0, 1]. One pitfall of using the above formula is that individuals with a constant response cannot be compared, as the Pearson correlation coefficient cannot be calculated when the variance is zero. In this special case, we consider two individuals with constant response to be completely similar to each other (returning a similarity value of 1). Thus, the phenotypic similarity measure is calculated using the formula:

$$\mathrm{PhenotypicSimilarity}(t_1, t_2) = \begin{cases} 1 & \text{if } \mathrm{Var}(t_1) = \mathrm{Var}(t_2) = 0 \\ 0 & \text{if } \mathrm{Var}(t_1) = 0 \text{ or } \mathrm{Var}(t_2) = 0 \\ R^2_{t_1,t_2} & \text{otherwise} \end{cases} \tag{1.5}$$

1.3 Test Setup

For analyzing the effects of selection mechanisms and algorithmic settings on GP population dynamics we ran test series using standard GP, GP with offspring selection and ALPS-GP. As benchmark data sets we used the *Poly-10* and the *Tower* data sets. For our test series we used the implementations of these algorithms and problems in HeuristicLab [14, Ver. 3.13], an open source framework for heuristic optimization that can be retrieved from http://dev.heuristiclab.com/.

The parameters for all here used algorithms were set such that they represent typical as well as competitive settings for the problem instances, i.e. typical configurations that are frequently used in practical applications and theoretical research studies.

1.3.1 Algorithms

1.3.1.1 Standard Genetic Programming (SGP)

First we applied symbolic regression using genetic programming as implemented in HeuristicLab. The following parameter settings were chosen for these tests:

- Population size: 500 individuals
- Termination criterion: 1000 generations
- Tree initialization: probabilistic Tree Creation (PTC2) [7]
- 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 [3]
- Terminal symbols: constant, weight * variable
- Function symbols: binary functions ($+, -, \times, \div$, exp, log)

1.3.1.2 Genetic Programming with Offspring Selection (OSGP)

Secondly, we used GP with strict offspring selection (OS) as explained in [1]. OS-GP 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 (i.e., 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:

$$SelectionPressure(i) = \frac{|GeneratedOffspring(i)|}{|SuccessfulOffspring(i)|} = \frac{|GeneratedOffspring(i)|}{|Population|}. \tag{1.6}$$

We use the selection pressure as termination criterion, i.e., 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 were set as follows:

- Population size: 200 individuals
- Termination criterion: Maximum selection pressure 200
- Parent selection: Gender specific [13]; proportional and random
- Offspring selection: Strict, i.e. success ratio = 1.0 and comparison factor = 1.0 [1]

1.3.1.3 ALPS GP

Age-layered population structure (ALPS) GP uses a novel measure of age to separate the population into multiple layers [4]. Each layer states a maximum age so that lower layers contain younger individuals and higher layers contain older individuals.

An individual's age determines how long it is allowed to remain in its current layer. Randomly generated individuals start with an age of zero, while in the default age-inheritance scheme, individuals generated by crossover inherit the age of the oldest parent plus one. Other inheritance schemes, such as using the younger or the average of the parents' age, have also been studied [5].

The age concept allows younger, less fit individuals to compete for survival within their own age layer, without being dominated by already matured individuals. A fair competition allows reseeding the lowest layer with new randomly generated individuals during the run, increasing the overall genetic diversity.

We used ALPS-GP as implemented in HeuristicLab and used the same operators and settings for tree size, initialization, crossover, and mutation as in standard GP and OS-GP. The following ALPS specific parameter settings were chosen for these tests:

- Population size: 300 individuals
- Age inheritance: Older
- Replacement strategy: Comma
- Aging: Age gap 20, polynomial aging scheme, i.e. the first layer (layer 0) is newly initialized every 20 generations, and individuals may move to upper layers at generations 20, 40, 80, and 160

1.3.2 Problem Instances

We tested the aforementioned GP algorithms on two benchmark regression problems to examine population dynamics. The problems were taken from the recommended GP benchmark problems [15] and are both available within the HeuristicLab framework.

- The *Poly-10* data set [8] consists of 500 samples with ten variables $x_{1...10}$ and the response variable y. The values $x_{1...10}$ 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 [12] 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 response 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 min. 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.

1.4 Test Results

Similarity and quality measurements were averaged over ten runs for each problem instance (*Poly-10* and *Tower*) and algorithmic configuration.

Figures 1.2 and 1.3 show the evolution of best and average quality and similarity values for standard GP. We notice that genotypic similarity remains at a constant level on both test problems. On the other hand, phenotypic similarity and average quality are higher on the *Tower* problem, suggesting a correlation between the two.

The distribution of similarity values per generation is displayed in Fig. 1.4a as 2d histograms, measured every 100 generations on the *Poly-10* problem. In the charts, the x-axis represents phenotypic similarity while the y-axis represents genotypic similarity. The results reveal that genotype similarity increases at a higher rate than the phenotypic similarity. The presence of multiple "islands" on the phenotypic similarity axis (at the same genotype similarity level) suggests that individuals in the population are organized into different semantic groups, some consisting of highly similar individuals.

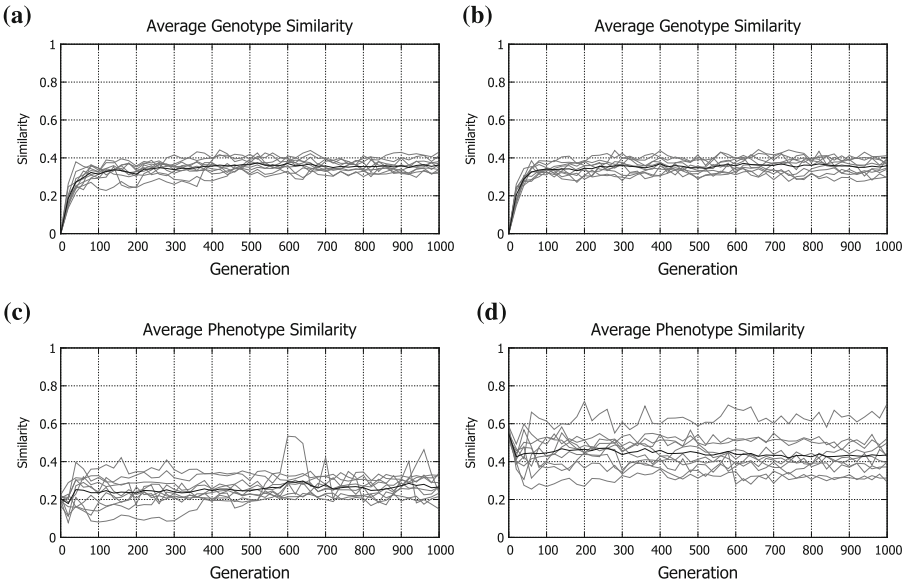

Fig. 1.2 Genotypic and phenotypic population diversity in standard GP, on the *Poly-10* and *Tower* problems. Thick lines represent average values over ten repetitions. (**a**) *Poly-10* genotypic similarity. (**b**) *Tower* genotypic similarity. (**c**) *Poly-10* phenotypic similarity. (**d**) *Tower* phenotypic similarity

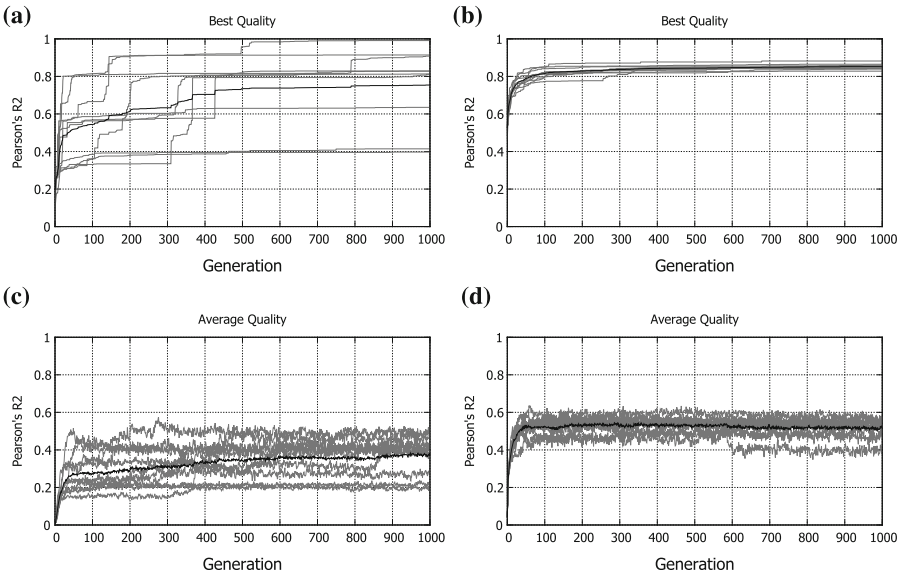

Fig. 1.3 Population quality in standard GP, on the *Poly-10* and *Tower* problems. (**a**) *Poly-10* best quality. (**b**) *Tower* best quality. (**c**) *Poly-10* average quality. (**d**) *Tower* average quality

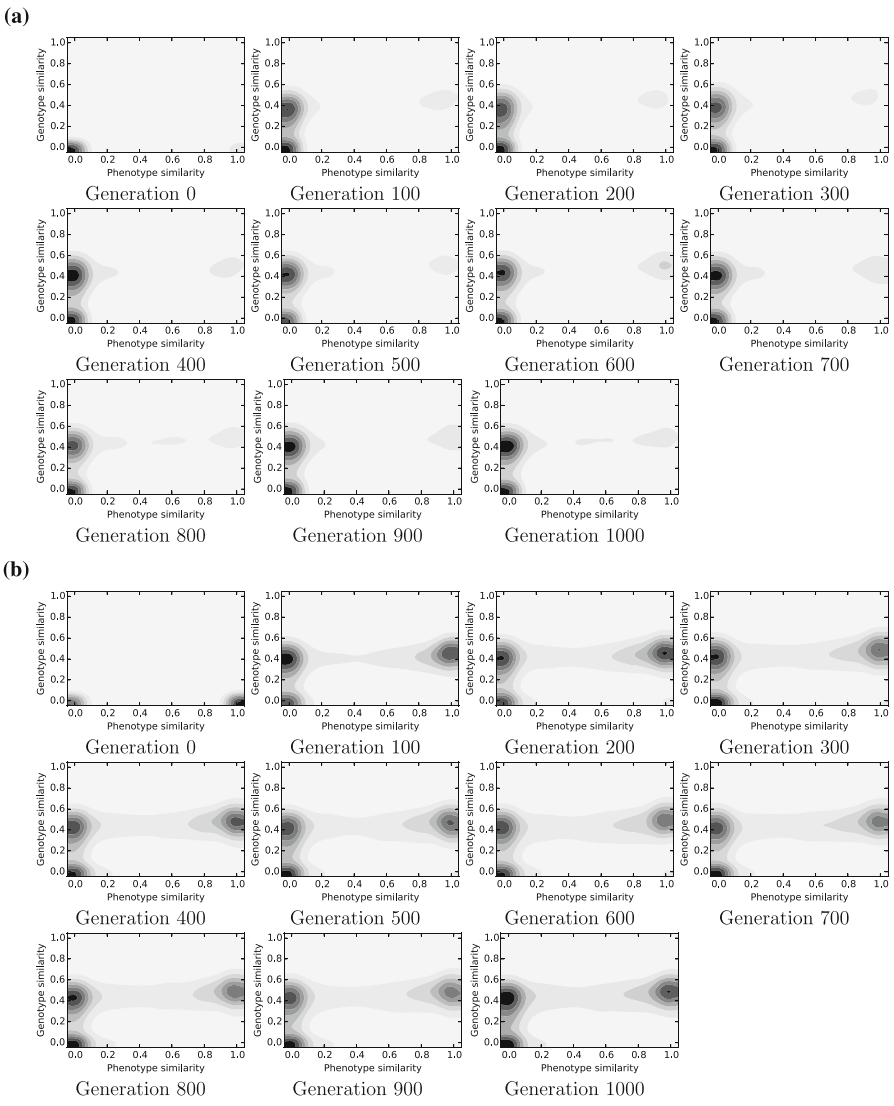

Fig. 1.4 Distribution of genotypic vs. phenotypic similarities in standard GP. (**a**) *Poly-10* problem. (**b**) *Tower* problem

We compare standard GP similarities with those measured on the OS-GP runs. Figure 1.5 indicates a steeper increase of similarity levels (both genotypic and phenotypic) towards significantly higher values.

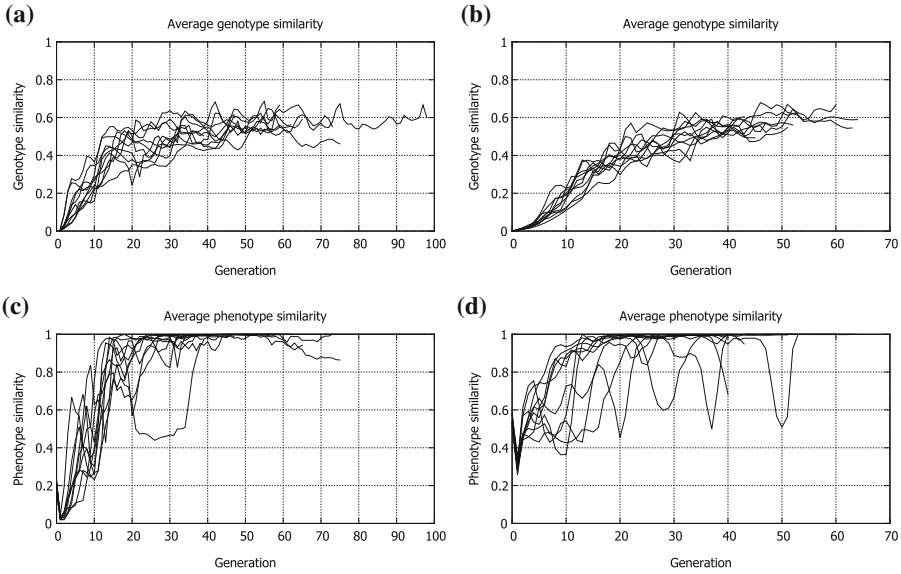

Fig. 1.5 Genotypic and phenotypic population diversity in OS-GP, on the *Poly-10* and *Tower* problems. (**a**) *Poly-10* genotypic similarity. (**b**) *Tower* genotypic similarity. (**c**) *Poly-10* phenotypic similarity. (**d**) *Tower* phenotypic similarity

The effects of strict offspring selection are particularly noticeable on the phenotypic similarity curves increasing asymptotically to a value of 1. At the same time, genotypic similarity is increased from an average value of about 0.35 (standard GP) to a value of approximately 0.5. Figure 1.6 shows average and best quality curves for OS-GP, which are almost identical due to strict offspring selection.

The distribution of similarities in Fig. 1.7 shows the movement of individuals in similarity space towards high genotypic and phenotypic similarity. We conclude that high semantic similarity heavily depends on the requirement that selection only accepts adaptive change (offspring with better fitness).

Figure 1.8 shows the overall average and best population quality for ALPS-GP, while Figs. 1.9 and 1.10 show the quality and similarity values per layer.

We notice that ALPS-GP is able to achieve a better average best quality than standard GP despite the fact that the overall average population quality is lower than the corresponding standard GP value, due to the lower-quality of the bottom ALPS layers. As expected, each age layer displays the same similarity behavior as standard GP, with the average population similarity increasing in the intervals