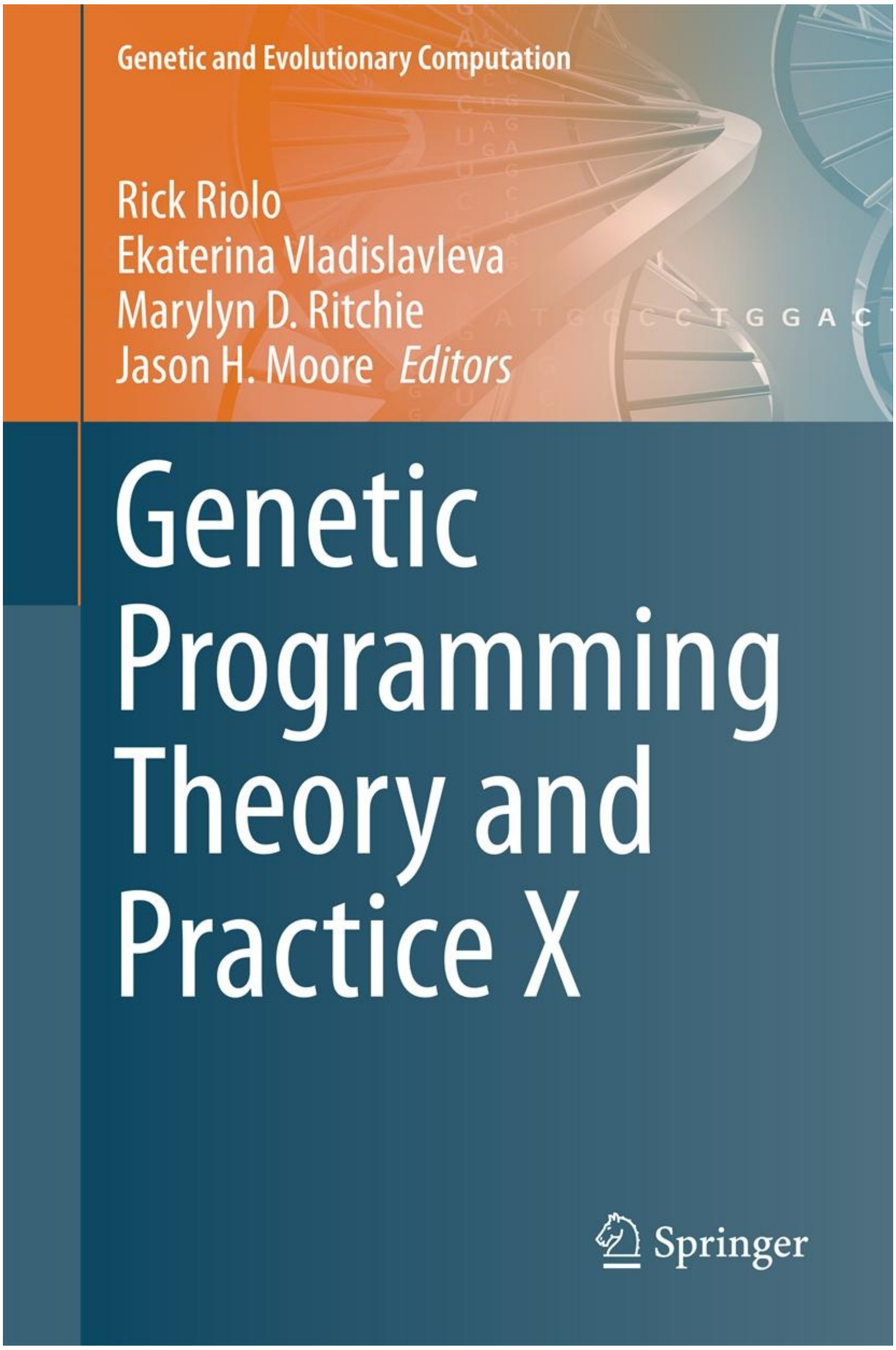
Genetic and Evolutionary Computation

Rick Riolo
Ekaterina Vladislavleva
Marylyn D. Ritchie
Jason H. Moore *Editors*

Genetic Programming Theory and Practice X

Springer


Genetic and Evolutionary Computation
Rick Riolo
Ekaterina Vladislavleva
Marylyn D. Ritchie
Jason H. Moore Editors
Genetic Programming Theory and Practice X
Springer

Rick Riolo, Ekaterina Vladislavleva, Marylyn D. Ritchie and Jason H. Moore (eds.), Genetic and Evolutionary Computation, *Genetic Programming Theory and Practice X*, 2013, DOI: 10.1007/978-1-4614-6846-2, © Springer Science+Business Media New York 2013

Genetic and Evolutionary Computation

For further volumes: http://www.springer.com/series/7373

Editors
Rick Riolo, Ekaterina Vladislavleva, Marylyn D. Ritchie and Jason H. Moore

Genetic Programming Theory and Practice X

Springer

Editors

Rick Riolo
Center for the Study of Complex Systems, University of Michigan, Ann Arbor, Michigan, USA

Ekaterina Vladislavleva
Evolved Analytics Europe BVBA, Beerse, Belgium

Marylyn D. Ritchie
Department of Biochemistry and Molecular Biology, The Pennsylvania State University, University Park, Pennsylvania, USA

Jason H. Moore
Institute for Quantitative Biomedical Sciences, Dartmouth Medical School, Lebanon, New Hampshire, USA

ISSN 1932-0167

ISBN 978-1-4614-6845-5 e-ISBN 978-1-4614-6846-2

Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013937720 such names are exempt from the relevant protective laws and regulations and therefore free for general use.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

This tenth anniversary edition of GPTP is dedicated to the memory of Jason Daida. Jason's presentations at the seminal GPTP workshops on structure and reachability inspired and greatly influenced our thinking and guided our research. Although his passion for teaching and education prevented his attendance at recent workshops, it was always a joy to encounter him be it at a conference or during one of many trips to UM's sister university in Shanghai. A quick and innovative mind coupled with a ready smile and positive outlook is a tough combination not to cherish. Jason's many students, friends and colleagues are testimony to his clear vision, dedication to learning and his love of life. We will miss him dearly.

Foreword

An Idiosyncratic Reflection on 10 Years of the Genetic Programming Theory and Practice Workshop

Beginnings

Ten years ago Carl Simon, then Director of the Program for the Study of Complex Systems (PSCS) at the University of Michigan invited me to lunch an asked me to give my input on a workshop on genetic programming (GP). Carl felt that as a growing, cutting edge field, it would be both useful and interesting for PSCS to sponsor a “state of the art” workshop of GP. As we discussed the idea, both Carl and I envisioned a one-time workshop that would bring together people actively working in the field. Little did I know that workshop would become a crucial part of my life and a regular event in my annual calendar.

At the time GP was still quite a young discipline despite 20 years or more effort on the part of many researchers. Carl was looking for a unifying theme for the workshop and after a few minutes of reflection, I suggested a theme of GP theory and practice, where computer scientists studying the theory of GP and practitioners applying GP to real world problems could meet and discuss their respective progress. It was my thought that such a meeting could provide a review of the current state of theory and that GP programmers could use a better understanding of GP theory to improve the application of GP to “real-world” problems. Conversely, practical results are the ultimate test of theory. Carl was enthusiastic about this idea and much to my surprise, asked me to work with Rick Riolo to organize the workshop.

Working with Rick was both a pleasure and an education. As I had never been involved in organizing an academic conference or workshop, I let Rick lead the way. Rick and the PSCS staff not only handled the logistics of the conference, but he knew the right questions to ask about format and content. We decided to try to have a matched pairing of theory and practice papers where possible, knowing that this would often be difficult. We also had long discussions about the format of the workshop. It was my idea that we should have longer times for presentations than was normal for conferences as well as plenty of time for discussion. We also decided that at the end of a set of related presentations, we should provide time for discussion reflecting on the set of presentations and what bigger questions they raised. These decisions have proved to be fruitful as many times the extended discussion sessions have been the most valuable part of the workshop.

Initially we conceived of the workshop as a place where people could present speculative ideas that they might not otherwise talk about at a peer reviewed conference. Instead, we opted for chapters to be written by presenters that were reviewed by other workshop participants and published in book form. While this meant that all attendees’ submissions would be accepted, they nevertheless went through serious review that often radically changed the chapter as did the lengthy discussion sessions during the workshop.

Another element we added was a daily keynote. Originally we planned for a generalized topic for a keynote speaker on each day: One day was to be keynoted by someone in evolutionary biology, one on evolutionary computing and one by someone who had expertise in integrating cutting edge technology into commercial applications. While this strict format has not survived, its spirit has survived and over the years the keynotes have spawned many fruitful discussions both during question-and-answer sessions after the keynote and in many discussions that extended late into the evening.

At the end of the first GPTP, it was by no means certain there would be a second workshop. It had been successful, but was not an unalloyed success in terms of content and quality. What was an overwhelming success was the interesting discussions at the workshop and deep into the nights at the end of each day. A little to my surprise, when asked whether they thought a second workshop was in order, there was an enthusiastically positive response from the attendees and from the entities that had provided financial support for the workshop, including the PSCS.

Over the years that have followed, the format has modulated somewhat, PSCS became a Center (CSCS) but the general ideas we settled on that first year, speculative presentations, diverse keynotes, large amounts of discussion time and cross-reviews by participants, have largely stayed intact. Moreover, over time the workshop has developed its own flavor and style that has led people to return; some annually, others biannually and still others only when they had something new to say.

Theory and Practice?

Perhaps the best way to describe the organizing principle of GPTP is the quotation attributed to Jan Schnapsheut (and Yogi Berra!) “In theory there is no difference between theory and practice. But in practice there is.” The first thing that quickly became apparent from the early GPTP workshops is that practice always outruns theory because it is much easier to think up a new scheme that helps to solve a problem but much less so to explain the mathematical reasons why such a scheme improves the fundamental function of the underlying algorithm.

The other thing that emerged was that practitioners became ersatz theorists, developing tools and metrics to test and explain behaviors in GP. Not only did this lead to modifications of existing algorithms and new techniques that were clearly shown to improve outcomes, but it spurred new theoretical consideration of GP. Theorists began to move from work on such fundamentals as the building block hypothesis to broader questions that approached some of the questions the evolutionary biologists wrestle with such as: What are the constraints on evolution? What are the dynamics? What are the information theoretical underpinnings of GP? There is also a growing sense that researchers in natural and artificial evolution have something to say to each other.

Selected Chapters

As the title of this forward suggests, I have an idiosyncratic view of GPTP. Approaching the 10th year, I decided to go back through the GPTP books published in past years, and pick some of my favorite chapters. This is totally subjective, with some of the chapters selected simply because they interested me personally, while others were chapters I selected because I thought they were particularly important to our improved understanding of GP and others, just because…What follows is the list of my choices from the first 10 years and some brief comments on them. This is by no means an exhaustive list or even a list of the "best" work done (but then evolution favors diversity over optimization), and I hope that people such as Trent McGonaghy, Erik Goodman and the many other people that I omitted from the list will not interpret this as lessening my respect for them or their work.

GPTP I: *"Three Fundamentals of the Biological Genetic Algorithm"* by Steven Freeland.

This keynote by the evolutionary biologist, Steven Freeland, outlined fundamental characteristics of natural evolution that he felt should be adopted by genetic programming. Some of the items he mentions include particulate genes, an adaptive genetic code, and the dichotomy between genotype and phenotype. He also sets a standard for measuring the success of evolutionary computing when he says "Biology will gain when evolutionary programmers place our system within their findings, illustrating the potential for biological inspiration from EC [Evolutionary Computing]."

GPTP II: *"The Role of Structure in Problem Solving by Computer"* by Jason Daida.

This chapter shows that there are natural limits on trees (and perhaps other related structures) that constrain the likely range of program-trees that can be created by standard genetic programming. This raises fundamental questions that have not been fully addressed in subsequent work.

GPTP III: *"Trivial Geography"* by Spector and Klein.

Spector and Klein showed that by creating a sense of place for individuals in a population and constraining their crossover partners to those in the near neighborhood, a significant improvement in efficiency and effectiveness can be realized. It also implicitly raises the question of an environment for evolution since once you have a sense of geography you can vary what is found in different locations (i.e., ecosystems).

GPTP IV: *"Pursuing the Pareto Paradigm: Tournaments, Algorithms and Ordinal Optimization"* by Kotanchek, Smits and Vladislavleva.

While the usefulness of Pareto optimization has long been recognized in evolutionary algorithms, this chapter was one of many chapters over many years by the authors that demonstrated that Pareto optimization is a key technique for effective genetic programming. Evolutionary programmers ignore it at their own risk.

GPTP V: *Towards an Information Theoretic Framework for Genetic Programming"* by Card and Mohan.

This is the beginning of a long and arduous journey by Stu Card and his associates to provide a model of genetic programming built on information theory. Now reaching its final, most general state, this may be the most important piece of theoretical work in the GP world yet. As a small joke, I once mentioned to Stu that since Lee Smolin proposed in his book The Life of the Cosmos that our universe evolved from earlier universes, Stu's work would be The Theory of Everything.

GPTP VI: *"A Population Based Study of Evolution"* by Almal, MacLean and Worzel.

This study done by my team imaged the dynamic changes of a GP population and demonstrated behaviors similar to those of natural populations, suggesting that GP behavior is closer to natural evolution than had previously been thought.

GPTP VII: *"Graph Structured Program Evolution: Evolution of Loop Structures"* by Shirakawa and Nagao.

I believe that using graph structures may lead to more powerful forms of GP and as an explicit structure altering technique, may overcome some of the limitations outlined by Daida in The Role of Structure in Problem Solving by Computer. While this chapter is fairly limited in its results, its method is powerful.

GPTP VIII: *"Genetic Programming of Finite Algebras"* by Spector et al.

This is not actually a chapter to be found in a GPTP book, being presented instead at GECCO in 2008, but Lee Spector presented this informally at GPTP-2009. It is an important paper in that he showed that GP was able to prove algebraic theorems that were too complex for human solution.

GPTP IX: *"Novelty Search and the Problem With Objective Functions"* by Lehman and Stanley.

This chapter is noteworthy if for no other reason, than because it calls into question the use of objective functions focused on accomplishing a specific result (even including the case of multi-objective functions). Instead it suggests that the search for novelty in GP derived programs may be more important, arguing that there is evidence in nature that novelty is more important than some hypothetical optimum. Moreover, it reinforces the argument that a more complex environment may yield better results.

GPTP X: *"A Practical Platform for On-Line Genetic Programming for Robotics"* by Soule and Heckendorn.

This was presented at GPTP-2012 by Terry Soule and will appear in the book you are holding (or reading online). It was built on a simple premise: Terry's group at the University of Idaho wanted to have a simple, easily programmable robot as a testbed for using GP in robotics. After looking at commercially available options for research robots, Terry concluded that there needed to be a less expensive, yet powerful and easily upgradeable platform as a testbed. They settled on a platform built from a number of off-the-shelf (OTS) components, with the computer being a smart phone. I include this both because I think it is an important tool for the GP community and because of the cleverness of how they assembled the components to make an inexpensive but powerful robot.

Thoughts on the Future of GP

Finally, as is typical after a review of the past, I want to take a guess at the future of GP, in the form of suggestions of desirable paths to be taken. As Alan Kay once said: "The best way to predict the future is to create it."

The GP community has a powerful opportunity to create the future as the continued growth of GP and its applications seems likely as the volume of data generated in all disciplines continues to grow. Methods such as GP, that can take data and turn it into information, will be of increasing importance. Of course, my suggestions of how we should approach the future are predictably biased by my experience and taste, so buyer beware!

The first area that seems ripe for further work is the growing collaboration between biologists and the GP community. Evolutionary biologists and evolutionary computer scientists not only share an interest in understanding the complexity of natural and computational evolution, but they also share a goal of building better models of complex processes. Some items where GP can build toward biology harken back to Steve Freeland’s keynote in 2003 where he recommended implementing a particulate gene model, diploidal chromosome structures and building more complex ecologies. All of these have been tried at one time or another in the history of GP, but I believe the time is right to produce a focused effort to build systems that integrate all of these elements.

On the flip side, deeper collaborations between the GP community and evolutionary theorists seems likely because of the growing use of computer models by biologists in all areas. The GP community can help in developing models by creating empirical models from biological data that can provide insight into first principles models that produce the data. Moreover GP tools can be used to image entire populations and model the dynamics of evolution.

The second area that I view as a rich area of exploration for the GP community is the question of what algorithms match the timescales of the systems being modeled and the possibility that GP could integrate different algorithms effectively. The point here is that in nature, evolution works on one timescale, ecology another and biology yet another. In machine learning techniques, neural nets work quickly, once they are trained. Artificial immune systems work on a longer timescale, responding somewhat more flexibly and evolutionary algorithms work on another timescale. I suspect that effectively integrating these different techniques may depend on recognizing the timescale on which they are most effective. It may also be possible to evolve an integrated solution using evolutionary algorithms to select component algorithms to solve larger computational problems with timescales as the constraint. I think this is particularly likely to be valuable in robotics, simulations and games (where many innovations first find a commercial home.)

Finally, I would like to call for GP to be applied to even more complex problems than has been the case to date. As our computing resources have continued to grow, and our improvement of fundamental algorithms and tools has progressed, it may be possible to address more difficult problems. Some areas may include symbolic proofs, complex problems such as the n-body problem or ecological models. The history of GPTP suggests that we may be at the point of pushing GP into more adventurous applications.

My view of the future of GP may be summed up by the following question: If the Singularity arrives, will it be by design or evolution?

And in Conclusion …

Some years ago, at one of the earliest GPTP workshops, Rick Riolo once described GP as “ … an art struggling to become a craft.” It is safe to say that with the modern tools and improved understanding of the GP mechanisms that has been generated in the last 10 years, it is at least a craft, and is beginning to be closer to an engineering discipline than ever before.

While it would be a gross exaggeration to say that this occurred because of the GPTP workshop, it is at least fair to say that GPTP has had a role in bringing together some of the best and most creative evolutionary engineers and theorists on an annual basis in a comfortable environment for 3 days of intense discussion, questions and speculation on an annual basis. I hope that the field will continue to mature and that the Genetic Programming Theory and Practice Workshop will continue as long as it continues to be useful.

In conclusion, I would like to thank the generosity of its supporters and, in particular, The University of Michigan and the Center for the Study of Complex Systems. In particular Rick Riolo’s role as midwife at GPTP’s birth and his quiet, steady role as parent for its growth is very much appreciated by all of the attendees over the years. Thanks Rick!

Milan, Michigan Bill Worzel

Bill Worzel
Milan, Michigan

Preface

Preface

The work described in this book was first presented at the Tenth Workshop on Genetic Programming, Theory and Practice, organized by the Center for the Study of Complex Systems at the University of Michigan, Ann Arbor, May 12–14, 2012. The goal of this workshop series is to promote the exchange of research results and ideas between those who focus on Genetic Programming (GP) theory and those who focus on the application of GP to various real-world problems. In order to facilitate these interactions, the number of talks and participants was small and the time for discussion was large. Further, participants were asked to review each other's chapters *before* the workshop. Those reviewer comments, as well as discussion at the workshop, are reflected in the chapters presented in this book. Additional information about the workshop, addendums to chapters, and a site for continuing discussions by participants and by others can be found at http://cscs.umich.edu/gptp-workshops/ .

The rest of this preface consists of two parts; (1) A brief summary of both the formal talks and of the informal talk during the scheduled and unscheduled discussions; and (2) acknowledgements of the many generous people and institutions who made the GPTP-2012 workshop possible by their financial and other support.

A Brief Summary of the Ideas from Talks and Talked About Ideas at GPTP-2012

As in the previous 10 springs, the 2012 workshop on Genetic Programming in Theory and Practice (GPTP) was hosted by the Center of the Study of Complex Systems of the University of Michigan. The discussions at the tenth jubilee gathering were particularly cohesive and friendly and nevertheless constructive, creative, and deep.

Hoping to repeat the success of the GPTP workshops of the previous years we planned lots of time for discussions and made the workshop longer. In 2012 it ran from Thursday morning till Saturday afternoon. Debates were full of open self-reflection, critical progress review and committed collaboration.

Thanks to our generous sponsors we could invite three keynote speakers this year and open every day of the workshop with an insightful and inspiring story. Thursday started with an address by Sean Luke on *“Multiagent Systems and Learning.”* Professor Luke, from the Department of Computer Science at George Mason University, has been an influential researcher is the fields of genetic programming and multiagent systems. His insight and experience in these areas contributed greatly to the workshop discussions about how to use genetic programming to solve complex problems. Friday began with a talk by Professor Seth Chandler on *“Evolving Binary decision trees that sound like law.”* Chandler, professor of Law at University of Houston, gave a remarkable and enlightening talk on applications of genetic programming in law. Not only did Professor Chandler show how to use genetic programming to evolve boolean expressions that predict the outcomes of legal cases and therefore sound like true “law”, he also provided a revealing comparison of GP-generated models with conventional approaches like decision trees, SVMs and NNs. His insightful illustrations of advantages of GP in terms of model compactness, transparency and interpretability as well as the unanticipated application area inspired many important discussions during and after the workshop. Saturday opened with Bill Worzel, then Chief Technology Officer of Everist Genomics, providing an "unkeynote address," *"A Random Walk through GP(TP)."* As Bill said "an unkeynote speaker will deliver a mostly retrospective talk, reflecting on what has happened, and perhaps a bit on why it has happened—call it the historian's view." Bill was present at and he was instrumental in the creation of GPTP and his perspective on the highlights from the past 10 years was educational, insightful, and entertaining.

Fifteen chapters were presented this year by newcomers and natives of GPTP on new and improved general purpose GP systems, analysis of problem and GP algorithm complexity, new variation paradigms, massively distributed GP, symbolic regression benchmarks, model analysis workflows and many exciting applications. The practice of GP was presented this year in a wide range of areas—robotics, image processing, bioinformatics and cancer prognostics, games, control algorithms design, stock trading, life sciences, and insurance law.

An important change this year compared with previous workshops was a more varied mix of different representations of GP individuals in presented systems. We made a coordinated effort to expand the topics of practical applications of GP far beyond GP symbolic regression for data fitting, and we think we achieved success.

Important topics in general purpose GP were the focus of many papers this year:

- Evolutionary constraints, relaxation of selection mechanisms, diversity preservation strategies, flexing fitness evaluation, evolution in dynamic environments, multi-objective and multi-modal selection (Spector, Chap. 1 ; Moore, Chap. 7 ; Hodjat, Chap. 5 ; Korns, Chap. 9 ; Kotanchek, Chap. 1 3);
- Evolution in dynamic environments (Soule, Chap. 2 ; Hodjat, Chap. 5);
- Foundations of evolvability (see Moore (Chap. 7) for co-evolution of variation operators, Giacobini (Chap. 4) for adaptive and self-adaptive mutation, Korns (Chap. 9), Flasch (Chap. 1 1) for parameter optimization);
- Foundations of injecting expert knowledge in evolutionary search (see Moore, Chap. 7 ; Benbassat and Sipper, Chap. 1 2; Hemberg, Chap. 1 5; Harding, Chap. 3);
- Analysis of problem difficulty and required GP algorithm complexity (Flasch (Chap. 1 1), albeit with empirical validation for symbolic regression); and
- Foundations in running GP on the cloud—communication, cooperation, flexible implementation, and ensemble methods (Babak, Chap. 5 ; Wagy, Chap. 6 ; McDermott, Chap. 1 4).

While GP symbolic regression was concerned with the same challenges as above, the additional focal points were:

- The need to guarantee convergence to solutions in the function discovery mode (Korns, Chap. 9);
- Issues on model validation (Castillo, Chap. 1 0);

- The need for model analysis workflows for insight generation based on generated GP solutions—model exploration, visualization, variable selection, dimensionality analysis (Moore, Chap. 7 ; Kotanchek, Chap. 1 3);
- Issues in combining different types of data (Ritchie, Chap. 8).

Another positive observation is that the existential discussions on whether GP can declare success as a science have dissipated from GPTP. The overall consensus is that GP has found it's niche as a capacious and flexible scientific discipline, attracting funding, students, and demonstrating measurable successes in business. Four companies using GP-based technology as their competitive advantage were represented among GPTP-2012 participants—Genetics Squared (cancer prognostics), Genetic Finance (stock trading), Evolved Analytics (plant and research analytics), and Machine Intelligence (image processing).

It looks like focus has shifted from being satisfied to generate beneficial comparisons of GP with other disciplines (e.g. GP symbolic regression with machine learning, see "do we have a machine learning envy?" in GPTP-2010) towards a more productive search for high-impact problems solvable with GP in various yet-to-be-conquered application areas, and massive popularization of GP.

An increasing gap between theory and practice of GP undoubtedly remains an issue. We doubt that this gap will ever be closed. Theoretical analysis of GP search performance in impossible without heavy constraints on the application area, representation, genotype-phenotype mapping, initialization, selection and variation mechanisms. First results were obtained last year for two simple problems (Neuman et al., 2011). The main challenge here is to make the analyzed problems as realistic as possible. The fact that all GP practitioners are aware of the countless number of small and big hacks that have made their GP algorithms considerably more effective adds to the staggering complexity of theoretical analysis of GP search. At this point in time the search for tight bounds on computational complexity for real problems seems intractable. We believe that attracting as many as possible hobbyists and interdisciplinary scientists to GP discipline, coupling research with other disciplines like fundamental computer science, mathematics, system biology, and a more systematic approach to GP can help bridge the gap between theory and practice.

Last year we stated that "symbolic regression and automated programming are just the two ends of a continuum of problems relevant for genetic programming: Symbolic Regression > Evolution of executable variable length structures > Automatic Programming. And while the 'simplest' application of GP to data fitting is well studied and reasonably understood, more effort must be put into problems where a solution is a computer program," (Vladislavleva et al., 2011). In response to this quest GPTP-2012 presented systems where GP individual was an sql-query (Spector, Chap. 1), an image filter (Harding, Chap. 3), a power control algorithm (Hemberg, Chap. 1 5), a game board evaluation function (Sipper, Chap. 1 2), a legal-case decision outcome (Chandler [1]), a stock-trading rule-set (Hodjat, Chap. 5), a robot micro-controller (Soule, Chap. 2), and a gene-expression classifier (Moore, Chap. 7). Such variety of representation could be an indication that we are slowly but steadily moving along the "Symbolic Regression > Evolution of executable variable length structures > Automatic Programming" path in the right direction.

We hope to solicit more work on evolving executable, variable length, structures in future workshops and facilitate understanding of missing mechanisms for using GP for automatic programing. GP shines in problems in which there is no single optimal solution is desired but rather a large set of alternative and competing local optima. Effective exploration of these optima in dynamic environments is perhaps the biggest strength of GP.

The idea to keep in mind are that many complex problems are modal and to solve them with GP we must relax selection mechanisms. How to do selection in a complicated dynamic environment where we never get enough information was, probably, one of the most popular questions at GPTP-2012.

- Thomas Helmuth and Lee Spector (Chap. 1) suggested that evolving programs with tags is one of the most expressive and evolvable ways to evolve modular programs, because tag matching implies inexact naming of individuals, and hence, more flexible selection.
- Soule (Chap. 2) addressed the problem of evolving cooperation and communication of robots online. He suggested that a hierarchical approach seems to be crucial for real-time learning at various time scales, and hierarchy is a form of niching. His chapter on designing inexpensive research robots to test on-board real-time evolutionary approaches has also contributed to another important goal addressed by many speakers at GPTP-2012—popularization of GP in other application areas, in this case—in robotics.
- Hodjat and Shahrzad (Chap. 5) proposed an age-varying fitness estimation function for distributed GP for problems where exact fitness estimation is unattainable, e.g. for building reliable stock trading strategies at long time scales.
- Harding et al. (Chap. 3) considered a flexible developmental representation—CGP to evolve impressive filters for object tracking in video using only limited set of training cases.
- Wagy et al. (Chap. 6) presented a flexible distributed GP system incorporating many relaxations to evaluation and selection mechanisms, e.g. data binning and island models.
- Moore et al. (Chap. 7) employed multi-objective Pareto-based selection with fitness and model size, as objectives in the computational evolution system for open-eded analysis of complex genetic diseases.
- Wagy et al. (Chap. 6) use an archive layering strategy as a means to maintain diversity in a massive scale GP system, EC-Star. Evolution here also takes a form of niching [2] where individuals are layered by a MasterFitness criterion, a kind of fidelity measure, reflecting the proportion of fitness cases against which individuals have been evaluated already.
- Korns (Chap. 9) presented complexity-accuracy selection niched per model age as a baseline GP symbolic regression algorithm.
- Flasch and Bartz-Beielstein (Chap. 1 1) provided empirical analysis of single-objective and relaxed multi objective selection for problems of increased complexity and demonstrated once again the undeniable advantages of niching

per complexity and age for more effective search in GP symbolic regression.

- Kotanchek et al. (Chap. 1 3) called for using as many competing objectives as possible, and varying them during the evolutionary search. The authors hypothesized that niching-based selection is the number one resolution for diversity preservation and effective exploration of complicated search spaces in dynamic environments.

When considering dynamic environments, inexact selection is directly related to issues of evolvability and open-ended evolution. The latter was addressed directly in several ways this year:

- Giacobini et al. (Chap. 4) introduced adaptive and self-adaptive mutation based of Levy flights as a flexible variation operator. Self-adaptive mutation is especially applicable to problems where length of evolutionary search is unknown upfront, and it is impossible to hardcode an optimal balance between exploration at the beginning of the search and exploitation towards the end. It seems that flexibly scaled massively distributed GP might benefit dramatically from the proposed self-adaptive mutation paradigm.
- Moore et al. (Chap. 7) have been facilitating evolvability and open-ended evolution by designed injection of expert knowledge into the evolutionary search.
- Benbassat et al. (Chap. 1 2) analyzed the same strategy of injecting domain knowledge for effective evolution of GP-based game players albeit with (naturally) less conclusive results. They discovered that for some games domain knowledge injection was definitely advantageous while for others not, illustrating the trade-off between flexibility (little domain knowledge) and specialization (a lot of domain specific knowledge).

Another topic related to evolvability is application of GP to problems with very different data sources. Ritchie et al. (Chap. 8) explored the problems with meta-dimensional analysis of phenotypes, the Analysis Tool for Heritable and Environmental Network Associations. The authors pled for solving issues with data integration in disease heritability research—the need for methods handling multiple data sources, multiple data types, and multiple data sets.

We were glad to witness once again the collaborative spirit of GPTP. Many open questions of GPTP-2011 were addressed this year. For example, the need for distributed evolution was answered in three chapters on GP system design targeted at massive distribution on a cloud (from 1,000 to 700,000 nodes) and generated a lot of debate. Island population model was considered to be one of the key strategies for flexible distributed evolution. However, McDermott et al. (Chap. 1 4) showed that the classical island model is not optimal for running GP on the cloud due to the lack of elasticity and robustness. The chapter raises insightful questions on the design of flexible evolution and provides initial experimental results comparing distributed and non-distributed design, flexible centralized vs. decentralized, vs. hard-coded, and static vs. dynamic population structure. Perhaps the most intriguing and arguably most applicable to elastic computation is decentralized dynamic heterogeneous GP design where population islands may differ in selection criteria, training data, GP primitives, the number of nodes can increase or decrease dynamically, and the system is robust toward communication failures between nodes.

Another design for a massive scale distributed GP system employing hub and spoke network topology is the EC-Star GP system presented by Wagy et al. (Chap. 6). The system is characterized by massive distribution capacity over come-and-go volunteer nodes, it's robustness, scalability and its particular applicability to time series problems with a extremely high number of fitness cases (e.g. in stock trading), when combined with age-fitness evaluation described in Hodjat et al. (Chap. 5).

From the general questions raised during discussions at GPTP-2012 we would like to distinguish the following:

- What are problems where solutions is a computer program? How to steer GP towards evolving programs?
- Can an algorithm evolved by GP learn during its execution?
- How to overcome the inherent problem of search space non-smoothness which emerges from the combination of representation and genetic operators? How to change the representations and variation mechanisms to allow minor adaptations? Is it necessary?
- How to optimally exploit and expand the concept of simple geographies?
- Maybe we should populate environments with subsets of training data?
- Should we pursue efficient strategies for parameter tuning or develop self-adaptive parameter servings?
- How to strike a balance between exploration and exploitation in open-ended evolution?
- How to seamlessly integrate different types of data structures?
- If the goal of many problems we are attempting to solve is understanding of underlying process, what are innovative post processing methods for analysis and final selection of GP solutions?
- Are diversity preservation and niching and expert knowledge sufficient for open-ended evolution?
- When solution accuracy is the goal, how to build self-correcting systems with built-in quality insurance?
- How to exploit modern architectures to run GP?
- How to characterize problems where either static or dynamic, centralized or de-centralized, homogenous or heterogeneous island models are beneficial for distributed GP?
- How many runs are enough to compare various algorithm setups?
- How to make hierarchical behavior in multi agent systems emerge rather than hard-code it?
- How to learn in general without too much reinforcement?
- How to enable supervised learning with very few training examples?
- How to do selection in environments where we never have enough information?

- What unites all methodologies we use for flexing the fitness evaluation and selection strategies?
- How to facilitate cultural propagation of GP to other disciplines? What is the strategy for bringing what we do to people who can benefit from it but do not know about it?

We are grateful to allsponsors and acknowledge the importance of their contributions to such an intellectually productive and regular event. The workshop is generously founded and sponsored by theUniversity of Michigan Center for the Study of Complex Systems (CSCS) and receives further funding from the following people and organizations: Michael Korns, John Koza ofThird Millenium, Babak Hodjat ofGenetic Finance LLC, Mark Kotanchek ofEvolved Analytics and Jason Moore of theComputational Genetics Laboratory of Dartmouth College.

We would like to thank all participants for another wonderful workshop. We believe GPTP do bring a systematic approach to understanding and advancing GP in theory and practice and look forward to the GPTP-2013.

Acknowledgments

We thank all the workshop participants for making the workshop an exciting and productive 3 days. In particular we thank the authors, without whose hard work and creative talents, neither the workshop nor the book would be possible. We also thank our three keynote speakers, Sean Luke, Seth Chandler and Bill Worzel.

The workshop received support from these sources:

- The Center for the Study of Complex Systems (CSCS);
- John Koza, Third Millennium Venture Capital Limited;
- Michael Korns;
- Mark Kotanchek, Evolved Analytics;
- Jason Moore, Computational Genetics Laboratory at Dartmouth College;
- Babak Hodjat and Genetic Finance LLC

We thank all of our sponsors for their kind and generous support for the workshop and GP research in general.

A number of people made key contributions to running the workshop and assisting the attendees while they were in Ann Arbor. Foremost among them was Susan Carpenter, who made GPTP X run smoothly with her diligent efforts before, during and after the workshop itself. After the workshop, many people provided invaluable assistance in producing this book. Special thanks go to Kadie Sanford, who did a wonderful job working with the authors, editors and publishers to get the book completed despite the many obstables, small and large. Courtney Clark and Melissa Fearon from Springer provided invaluable advice and editorial efforts, from the initial plans for the book through its final publication. Thanks also to Springer’s Latex support team for helping with various technical publishing issues.

References