Chapter 1

# GENETIC PROGRAMMING: THEORY AND PRACTICE

## *An Introduction to Volume III*

Tina Yu,[1] Rick Riolo[2] and Bill Worzel[3]
[1]*Chevron Information Technology Company,* [2]*Center for the Study of Complex Systems, University of Michigan,* [3]*Genetics Squared, Inc.*

*In theory, there is no difference between theory and practice. But, in practice, there is.*
—Jan L.a. Van De Snepscheut

**Keywords:** genetic programming, theory, practice, continuous recurrent neural networks, evolving robots, swarm agents

## Close Encounter, the Third Time

To leverage theoretical and practical works in the field of genetic programming (GP), the Genetic Programming Theory and Practice (GPTP) Workshop series was conceived and launched in 2003. For the past two years, theoreticians and practitioners have come to Ann Arbor to present their works and to listen to others' (Riolo and Worzel, 2003) (O'Reilly et al., 2004). Gathered in a friendly environment, they debated with enthusiasm, pondered in silence, and laughed in between. All of these interactions have paved the way to future integration of theory and practice.

In this year's workshop, we are very pleased to see some signs of convergence:

- Papers developing techniques tested on small-scale problems include discussion of how to apply those techniques to real-world problems, while papers tackling real-world problems have employed techniques developed from theoretical work to gain insights.

- Multiple papers addressed GP open challenges, such as industry funding, new opportunities and previously overlooked issues. During the open discussion on the last day of the workshop, considerable enthusiasm was generated regarding these topics.

All those developments indicate that both theoreticians and practitioners acknowledge that their approaches complement each other. Together, they advance GP technology.

## 1. Three Challenging Keynote Talks

As in the first two GPTP workshops, each day commences with a keynote talk from a distinguished researcher, one each with a strong background in the fields of evolutionary computation, biology and application of advanced technologies in real-world settings, respectively. For GPTP-2005 we were again fortunuate to have three enlightening, inspiring, challenging and sometimes controversial talks.

On the first day of the workshop, Van Parunak, Chief Scientist of Altarum Institute, delivered a keynote on evolving "Swarms" of agents in real-time. As a practitioner of population-based search techniques, one of Van's challenges is mapping a real-world problem into an appropriate representation. Sometimes, each individual in the population is the entire solution while other times, an individual is one component (an agent) of a solution. In the later case, the collection of individuals (the "Swarm") which yields the desired global behavior is the solution. The art and craft of designing problem-specific representations mentioned by Van was a challenge echoed by other presenters throughout the workshop.

One type of real-world problem that Van works on is to evolve swarms in real-time to meet a constantly changing environment. In Chapter 2, he discusses two such systems they have developed. The first one plans flight paths for uninhabited robotic vehicles (URVs). The path should lead URVs to the target while avoiding threats on the way. To detect moving threats, an URV generates many "ghost" agents which explore (in a virtual model of the world) possible paths by depositing digital pheromones. Each step in the path then is chosen based on information represented by the pheromone deposits, using a parameterized equation associated with the ghost agent. The Altarum group has explored several approaches to optimizing the parameters in real-time to guide URVs, including evolutionary algorithms and human designers. The evolved parameters produce paths that are superior to those produced by human designed parameters by an order of magnitude.

Using the ghost agent concept, they developed a second system to predict future behavior of soldiers in urban combat. A soldier's behaviors are influenced by his/her own personality, the behaviors of other soldiers and their surrounding environment. To extrapolate a soldier's possible future behavior, a stream of ghost agents are continuously generated. These ghost agents begin their lives in the past using a faster clock than the clock used by the soldier it represents. When the time reaches the present, the ghost agents whose behaviors match well with the past behaviors of the soldier it represents are assigned a high fitness. These ghost agents are allowed to bred offspring and to run past the present into the future, where their behaviors are observed to derive predictions.

Modeling complex systems in real-time, with models that run and adapt faster than real-time in order to allow for prediction, is a non-trivial task. Van showed us one way to make it work. However, he acknowledged that their efforts were aimed at solving the problems at hand, and hence so far they have not focused on generating theoretical insights. However, he asserts that although the systems they have developed doesn't give "perfect" predictions, it outperforms the current systems in use. From the practical point of view, it is a success. This evaluation standard is also used in other lines of business, such as finance, chemical and oil companies, as confirmed by the work and comments of other workshop participants.

The second day started with a keynote entitled "Evolution From Random Sequences" by Mike Yarus, Professor of Molecular Biology at University of Colorado, Boulder. This is not evolution by mutation of existing sequences with a fixed translation mechanism generating "solutions," he emphasized. Instead, it is a completely different process where both the genetic code (information) and the translation system (a "machine") are randomly generated, and evolution proceeds as selection acts upon this coupled pair.

Their studies are based on the laboratory examination of the RNA-binding sites of eight biological amino acids, which show significant evidence that cognate codons and/or anticodons are unexpectedly frequent at these binding sites. Consequently, they proposed the *Escaped triplet theory*: The coding triplets began as parts of amino acid binding sites, then escaped to become codons and anticodons. In other words, at least part of the genetic code is stereo-chemical in origin-from chemical interactions between amino acids and RNA-like polymers. The code is not just a *frozen accident* as suggested by Watson and Crick. Instead, the code's mapping is a result of selection based on affinities between an amino acids and parts of random RNA sequences.

Not only the genetic code is selected from random sequences, Yargus argued—so is the hardware for translation. He used the peptide transferase to support his argument. Their laboratory study shows that proteins are assembled by reaction of the aa-RNAs within a cradle of RNA whose octamer can be selected from random sequence. Therefore, both coding triples and the peptidyl transferase emerge when random sequences are placed under selection. Put another way, they were originally made by selection from populations of RNAs of arbitrary sequence.

The issues involved with the *invention of a genetic code* are generally not considered by the GP community, who usually assume the existence of a "code" and machinery to map from a "genome" to active agents (*e.g.*, programs). However, as a field constantly looking to biological mechanisms and processes for inspiration, GP might due well to consider these issues in the future, perhaps leading to more "open-ended" evolutionary systems.

Following a suggestion to be challenging and controversial, Inman Harvey delivered a keynote on "Evolutionary Robotics for Both Engineering and Science" with comments on some aspects of GP and the interaction of human and evolution process. He started by describing their approach to evolve dynamic systems which interact with the environment in real-time. Formally, a standard dynamic system is a set of (continuous) variables with equations that determine how each variable changes over time as a function of all current values. These equations are represented in Continuous Time Recurrent Neural Networks (CTRNN) and are evolved using a steady-state GA with tournament selection.

Inman was questioned about his decision to not use GP for the evolutionary component. He gave his reasons based on his observations of the early GP work. First, he thought GP-style evolution is *wide and short*, *i.e.* it consists of a large population evolving for just a few (*e.g.*, hundreds or fewer) generations. But biological evolution is *narrow and long*, *i.e.* the number of generations is generally far more than the size of the population. Secondly, biological evolution is always an open-ended work in progress, not just an attempt to solve a single specific problem. It seemd likely that Inman has not been in touch with the GP field for a long time and thus he did not have much familiarity with recent progress and trends. Workshop participants quickly corrected his misconceptions, claiming that those ideas have been incorporated in some of the more current GP systems. However, Inman's basic point should still be seriously considered, *i.e.*, while GP systems are run longer and are work toward more openedness than in the past, it is clear that the ratio of generations to population size is still far from that in biological systems, and that GP systems are still generally applied to solve specific problems. It then remains to be seen how important those differences are across the range of GP applications, given the different goals researchers have for GP systems,

The subject then turned to the evolutionary robotics (ER) systems Inman's group has built for scientific purposes. The first one is an artificial ant that has to find its way back to its nest or hive with minimal noisy visual cues. Biologists used the system to compare simulation behaviors with the real ant behaviors to disprove or to generalize hypothesis. For example, if the original hypothesis states that a behavior requires $A$ and the evolved artificial ant show the behavior without $A$, a new hypotheses can be developed to explain this behavior. Another ER system they developed is for studying the human ability to adjust to a world turned upside-down. They incorporated some general homeostasis constraints to evolve a robot with normal eyes first. After that, they switched the eyes upside-down and ran the system again. A reasonable proportion ($\tilde{5}0\%$) of the evolved robots with normal eyes can adapt, after time, to visual inversion. These experiments allow generation of relatively unbiased models (*i.e.*, with minimal assumptions) to challenge existing hypotheses and to generate new ones.

For engineering purposes, Inman and his group applied their ER technique to evolve control systems for robots. Two such examples are a hexapod walker for a robot for Mars exploration that is robust to damage and a humanoid biped walker. They used an incremental approach to evolve the system. Initially, a hand-designed system for a simple task is used at population 0. Once the evolved system is able to perform the simple task reasonably well, a new task (parameters and neurons) is added and starts a new evolutionary cycle. Evolution gradually learns to perform new tasks without forgetting how to do the old task. This style of incremental learning through the interaction of human intervention and an evolutionary algorithm is a practical approach to tackle this engineering task. However, it seems to conflict with the *work in progress* evolutionary paradigm that Inman advocated previously, pointed out by a workshop participants. Inman agreed with this comment. Maybe devising an evolutionary system which can continuously learn, *i.e.* always in work-in-progress mode, without human intervention is a challenge for all who are interesting in evolutionary learning, not just those using GP.

## 2. Real-World Application Success Stories

Besides the successful applications of evolutionary approaches described by Van Parunak and Inman Harvey in their keynote addresses, clear-cut Genetic Programming success stories were told in four presentations. They either produced better results than the preexisting systems, made breakthroughs or opened a new frontier. These results cheered the spirits of all workshop participants.

In Chapter 3, Lee Jones, Sameer H. Al-Sakran and John Koza present their success in delivering GP human-competitive results in a new domain: optical design. In this work, the simple forms of representation, genetic operations and fitness function were elaborated to work with this non-trivial domain, where finding a solution is an art or craft rather than science. Many pathological designs were identified and the system was adjusted accordingly to avoid generating such kinds of designs. As an invention machine, GP was able to create lens designs that gives characteristics, *e.g.* spherical aberration and distortion, that are competitve with a lens design patented in 1996. Since the evolved design differs considerably from the patented design, it does not infringe the patent. Instead, it is considered as a new invention created by GP.

Chapter 4 also reports the success of a GP solution that improves over a preexisting technology. In this work, Frank Francone, Larry Deschaine, Tom Battenhouse and Jeffery Warren applied a linear GP system to discriminate unexploded Ordnance (UXO) from clutter (scrap metal that poses no danger to the public) in retired military fields. A higher quality solution allows UXO to be revealed by digging fewer holes, hence is more cost-effective. The project was conducted in two phases. The first phase used sensor data gathered from a military field where UXO and clutter locations are known. The quality of a solution is evaluated by the percentage of UXO and clutter correctly identified. They compared the GP-generated solution with solutions based on geophysics first principles and by other technologies, and showed that the GP-generated solution gives a significantly higher accuracy. In the second phase of the project, the sensor data was collected from a different field where UXO and clutter locations are unknown. In order to devise GP solutions, many more processing steps, such as anomaly identification and feature extraction for the identified targets, were conducted. Unlike the phase I study, the quality of a solution in this phase is judged by the number of holes that must be dug to uncover all UXO. They reported that their GP-generated solution improves over the preexisting technique with 62% fewer holes dug. Although the data set is noisy with only a small number of positive samples, a common dilemma in real-world applications, GP is able to overcome the difficulties and deliver good solutions.

In last year's workshop, Lohn, Hornby and Linden presented their success in evolving two human-competitive antennas for NASA's Space Technology 5 mission. While those antennas met the mission requirements at that time, new requirements were introduced as a result of an orbit change. In Chapter 5, they updated the project with two new antennas they evolved to meet the new mission requirements. Unlike the conventionally designed quadrifilar antenna which require several months to develop a new design and prototype it, their antennas were evolved (with slightly modifications of their evolutionary system) and prototyped in four weeks. These two antennas have passed the flight testing and are expected to be launched into space in 2006, a "first" for systems designed by evolutionary algorithms. This story highlights an important advantage of evolutionary design over human design: the ability to rapidly re-evolve new designs to meet changing requirements. It is an essential ingredient for successful real-world applications.

Variable selection plays an important role in industrial data modeling, particularly in chemical process domain where the number of sensor readings is normally large. To generate robust models, a small number of important variables must be identified. Unfortunately, preexisting linear variable selection methods, such as Principle Components Analysis (PCA) combined with Partial Least Squared (PLS), fail to work on non-linear problems. In Chapter 6, Guido Smits, Arthur Kordon, Katherine Vladishlavleva, Elsa Jordaan and Mark Kotanchek developed a non-linear variable selection method based on their Pareto GP system. This method assigns variable importance by evenly distributing an individual's fitness to all variables that appear in the individual. The accumulated importance of each variable in the population in the Pareto front archive is then used to rank their importance.

They have applied this method on two inferential sensors problems. The first one (emission prediction) has 8 variables and GP selected 4 of them as highly important while PCA-PLS gives a different ranking. The final deployed models, which were evolved by GP using the 4 selected variables, give very high correlation coefficient values (0.93 and 0.94). This confirms that the 4 selected variables are indeed important, which PCA-PLS fails to recognize. The second inferential sensor (propylene concentration predication) has 23 variables. Four important variables were selected by GP whereas PCA-PLS suggests 12 important variables, which included only 3 of the 4 GP selected variables. The final winning inferential model is an ensemble of 4 models, which included all 4 GP-selected variables and 1 variable recommended by an expert's model. The GP solution also was more effective than the PCA-PLS solution in this case.

In addition to providing demonstrably better performace, one prerequisite for "success" is acceptance by the people working in the problem domain. It is only when the solutions are accepted by the users in the domain that the technology will have a significant impact. Thus an important question is: Are those fields where GP has been applied inclined to accept the solutions? If not, how do we change their attitudes?

The feeling of the GPTP Workshop participants was that in general, the more successful and mature a field is, the less likely it accepts new ideas. Lens and analog circuit designs are two fields that have longer histories and are considered more mature, said Koza. In contrast, antenna design engineers and geophysicists working on UXO communities are very accepting of new concepts as there is not solid theory and they don't know systematic approaches for finding solutions themselves, according to Lohn and Francone. In terms of enticing end-users to accept GP solutions, one critical step is to invite them to participate in the project from the very beginning, said Kordon. Otherwise, people tend to not accept any work that they have no part of. In corporate environments, it also is important to show management the advantages the technology can bring to them. If the success of a technology will lead to problems for them, *e.g.* losing their jobs, they will make every effort to assure the technology fails, commented by Goodman.

## 3. Techniques with Real-World Applications in Mind

Although GP theory does not progress as rapidly as practice does, techniques to enhance GP capabilities and theoretical work to analyze GP processes are continually being developed. Four such papers were presented in the workshop. These works so far have been applied to small scale problems. Nevertheless, relevance to real-world applications was discussed.

In Chapter 7, Tina Yu introduced a functional technique to evolve recursive programs. In functional programs, recursion is carried out by non-recursive application of a higher-order function. This chapter demonstrates one way to evolve this style of recursive programs by including higher-order functions in the GP function set. Two small-scale problems were studied using this approach. The first one is a challenge by Inman Harvey, STRSTR C library function, and the second one is the Fibonacci sequence. In both cases, problem-specific knowledge was used to design/select higher-order functions, and GP was able to evolve the recursive programs successfully by evaluating a small number of programs.

Programs with higher-order functions naturally give the structure of code abstraction and reuse. For these two problems studied, the structures were defined by the given higher-order functions. With an appropriate set-up, GP can be used to discover the structure, *i.e.* evolve the higher-order function. Such a GP would be particularly suitable for solving open-ended designs where no optimum is known and creativity is essential to problem solving. In this case, evolved higher-order functions might deliver interesting solutions.

Lee Spector and Jon Kleinsold present their "trivial geography" technique in Chapter 8. Trivial geography structures the GP population in a simple geographically distributed manner. The location of an individual is taken into account when selection for competition and reproduction. This concept is not new. Many existing evolutionary computation systems divide their populations into discrete or overlapping sub-populations, often called demes, as a form of geography. However, their implementation is significantly simpler; only a few lines of programming code need to be added/modified, they argued. In their implementation, a population is structured as a ring. When producing a new generation, the location into which an offspring is going to be placed in the new population decides where its parents are from; *i.e.*, only the individuals near to the location for the offspring are selected for tournament and thus are candidates to be parents. This essentially gives overlapping sub-populations where independent evolution takes place. Despite being such small change, this trivial geographic bias in parent selection significantly improves performance for the two problems they tested. Although the generality of the method has not been studied yet, they recommended broader usage of the technique. "It is easy to implement and you might be surprised what you can gain from it," said Lee.

In Chapter 9, Riccardo Poli and Bill Langdon developed a backward chaining technique to reduce GP computational efforts. This technique first reorders the typical create-select-evaluate evolutionary system cycle to construct the genealogy network for the entire evolutionary run. After that, the genetic makeup of the individuals are filled in a backward manner. This is done by tracing the genealogy of each individual in the last population back to generation 0. The "root individuals" are then initialized randomly and all their descendants are created using genetic operators subsequently. Since only individuals in the geneological network are created and evaluated, backward chaining GP is computationally more effective than the traditional GP. However, there is trade-off of memory to store the genealogy network. Mathematically, they computed the time and space complexities to show the cost and saving. Experimentally, they tested this technique on symbolic regression problems and reported that using population size 10000 with tournament size 2, backward chaining GP gives computational saving of 19.9%. Once the tournament size is increased to 3, the saving is marginal. They recommend this method to GP systems with very large populations, short runs and relatively small tournament sizes. The computational saving for large scale real-world problem using this type of GP might be significant.

Co-evolving grammar and the solutions defined by the grammar is an attractive idea since the biases induced by the grammar are not always favorable throughout the evolutionary run. Conceptually, it seems that it should be possible to learn good bias from the evolved good solutions. In Chapter 10, R. Muhammad Atif Azad and Conor Ryan test the hypothesis by using a diploid genotype: one part for the grammar rule and the other for solution mapped. This approach is very similar to the co-evolution of genetic operation rates and the solutions generated by the operation. By encoding the rate as a part of the genotype, the rate is normally reduced as evolution progresses to provide appropriate exploration and exploitation.

They added the diploid genotype to their Grammatical Evolution system and tested it on a set of small scale problems. While the results are not as good as expected—the system using static grammars finds better solutions—this talk stimulated much discussion at the workshop. Many recommendations were given to improve the system.

Chapter 11 is a contribution by Tuan Hao, Xuan Nguyen, Bob McKay and Daryl Essam. This work applies their previously developed techniques to a real-world problem, which is an important step to transfer the technology for wider applications (Bob was not able to come to present the paper in person, so there was not discussion of it at the workshop). Their work is based on Tree Adjoining Grammar (TAG) GP which they have developed and used to study two local search operators: point insertion and deletion. Local search operators are generally useful to tune final solutions. While their previous study reported that they are also effective search engines on small-scale problems, when applied to the larger scale ecological modeling problem described in Chapter 11, the results are not conclusive. On training data, GP with local search operators produces a better model than the model evolved by GP alone. However, on blind testing data, it is the other way around. This indicates that local search operators generate over-fitting solutions and reduce generality. They are continuing the study to produce more robust solutions.

## 4. Visualization: A Practical Way to Understand GP Process

Unlike the work describe by Mike Yarus in Section 1, which examines biological data to study evolution, A. Almal, W. P. Worzel, E. A. Wollesen and C. D. MacLean analyze biomedical data for diagnostics and prognostics purposes. One such project is modeling medical data to predict the stage of bladder cancer. Medical data is notorious in its small sample sets and large dimensionality, which makes the modeling task very difficult. In Chapter 12, they describe a tool to visualize the content diversity (the diversity of functions and terminals) of GP populations and study its relationship to the fitness diversity of the solutions.

They used the new tool they developed to plot population contents in generation 0, 10, 20 and 38, which show how diversity decreases as evolution progress. Fitness diversity, however, does not have such a trend. The fitness variance among individuals remained high throughout the runs, although high fitness bands became dominant when the content diversity became very low, *i.e.*, the population's structures converged. This interesting relationship stimulated much discussion at the workshop. The relationship between structure, content and fitness in a population is a subject that always interests both theoreticians and practitioners.

Visualization is a powerful and practical way to study many dynamical systems, including those generated by evolutionary processes. Thus, it may not be surprising that there were three other visualization papers presented at the workshop.

The first one is by Christian Jacob and Ian Burleigh. In Chapter 13, they present an agent-based model that simulates lactose operon gene regulatory system. Although this is one of the most extensively studied biological systems, there are still many unknowns. A visual simulation can help biologists to understand the complex system better. To develop such a model, they first incorporated biological data/rules to construct the system. The simulation behaviors are then presented to biologists, whose feedbacks are used to improve the model. This interactive evolution process led to parameters which give behaviors close to the known behaviors. It appears that GP can be used to fine-tune the parameters. Furthermore, the mechanism of the gene regulatory system may serve as an inspirational platform to design GP systems suitable for complex systems modeling.

Biological systems have always been inspiration to GP. Motivated by the research of neutral networks in biological systems, Wolfgang Banzhaf and Andre Leier investigate GP search behavior in a Boolean function space with the presence of neutral networks. In Chapter 14, they enumerated the problem search space and showed that the genotype to phenotype mapping is similar to the RNA folding landscape: there are many very uncommon phenotypes and few highly common phenotypes. This suggests that the neutral evolution theory for biological systems might apply to this GP search space. They plotted the phenotype network of the search space, including neutral networks where the connected phenotypes having the same fitness. This visualization of the network provides a clear picture of phenotypes with different fitness and how they are connected.

Another work which relies heavily on visualization for analysis is by Ellery Crane and Nic McPhee. In Chapter 15, they study the effects that size and depth limits have on the dynamics of tree-based GP. Based on a simple one-than-zero problem, many GP experiments were conducted using both tree-size and depth-size limits. Visualization of the statistical results indicates that both kinds of limit have similar effects on the average tree size (number of nodes) in the population. However, depth limits effect program shapes more than size limits do. With depth limits, the program shape in the population has less diversity. They are investigating the generality of this phenomena by studying other type of problems under different selection and genetic operation conditions, and if practitioners adopt their recommendations for problem solving, we may learn even more about its generality and usefulness.

## 5. Open Challenges

In addition to the deep challenges presented by the keynote addresses, several other chapters also described various kinds of open challenges that GP practitioners must overcome before GP will be easily and widely accepted in various industries and business.

For example, in Chapter 16 Arthur Kordon, Flor Castillo, Guido Smits and Mark Kotanchek of Dow Chemical discuss many challenges faced by industrial research and development groups when applying GP technology. In addition to technical issues, such as data quality and extrapolation of the solutions, non-technical issues are important to the success adoption of a new technology in corporate environment. They summarized how they address these non-technical issues: create a team to work on GP, link GP to proper corporate initiatives, secure management support, address skepticism and resistance and marketing the technology continuously. Although GP has had good track record at Dow, the technical team still has to adapt to the fast changing environment and to produce profits to survive. They described a set of "10 commandments" of industrial R&D humorously to illustrate the challenges they are facing:

- Thou shalt have no other thoughts before profit.
- Thou shalt not serve consultants.
- Thou shalt not take the research money for granted.
- Remember the budget, to keep it holy.
- Honour the cuts and the spending targets.
- Thou shalt not explore.
- Thou shalt not commit curiosity.
- Thou shalt not create.
- Thou shalt not develop anything before outsourcing it.
- Thou shalt not covet thy professors, nor their students, nor their graduate students, nor their post-docs, nor their conferences and workshops.

Open-ended problem solving has been a quintessentially human capability. Is it possible to equip GP to become the first machine capable of open-ended problem solving? In Chapter 17, Jason M. Daida argued that it would be very difficult, if not impossible, based on the MPS open-ended problem solving paradigm. In this widedly used problem solving paradigm, there are 6 stages of problem solving: engage, define stated problem, create internal idea of problem, plan a solution, carry out the plan and evaluate (check) and look back. Clearly, it would be very hard for GP to undertake some of the activities, *e.g.* engage and define stated problems. In fact, until now, GP has been partnered with human to carry out these problem solving activities. This is demonstrated in typical GP application work-flow, which includes pre-GP (*e.g.* data preparation) and post-GP (*e.g.* solution interpretation) process. Nevertheless, there are opportunities to make GP a more competent partner. One such area is tools to transform/analyze GP solutions so that they can be explained and incorporated into the evaluate, check and look back process. Visualization has been recommended as one great approach to achieve the goal. There are many other opportunities to strengthen GP which remains open for the community to explore.

Jianjun Hu, Ronald Rosenberg and Erik Goodman have started exploring new application domains using their bound-graph representation GP system. Chapter 18 reports their initial study on evolving mechanical vibration absorbers.

This is an area with a history of patents and it poses a great challenge for GP human-competitive results. To evolve single, dual and bandpass vibration absorber, they designed various domain-specific functions. They also devised different fitness functions to direct GP search. The evolved absorber, however, are not practically useful and extremely difficult to implement, although their fitness are high. They concluded that exploiting domain or problem-specific knowledge to embody physically meaningful building blocks is necessary for GP to be successful in real-world problems. Otherwise, the evolved solutions may not be physically realizable. How much domain knowledge to use so that GP has room for creativity and is able to deliver human-competitive results is an open challenge for the community.

Pushing GP toward industrial success in the analog CAD domain, Trent McConaghy and Georges Gielen outline new GP applications and challenges in Chapter 19. They started by distinguishing "success" in the GP research domain, which is demonstrated by the number of publications, and in the industrial success, which is measured by the number of different chip designs that have been sent to fabrication. With great research success in analog design, they suggested using GP to pursue industrial success in three application areas: automated topological design, symbolic modeling and behavioral modeling. They showed their recent work on these problems. The results are very encouraging and accepted well by the CAD design community. Although there are many obstacles to overcome, *e.g.* computational feasibility and earning CAD designers' trust, these applications are great opportunities for GP to become industrial success in the analog CAD field.

There was a lot of interest in discussing GP challenges throughout the workshop. On the last day, a list of open challenges was created by workshop participants:

- Handling large data sets (10 millions).
- Complexity of problems (k-complexity).
- How weird can GP be and still be invited to GPTP?
- The problems associated with analysis of GP systems.
- Mapping GP to customer satisfaction.
- How do we stack GP techniques (avoid "backdrop").
- GP integration with other techniques.
- Theoretical tools for understanding large modular systems.
- How do ADFs affect the GP system?

- Systematizing our understanding of GP: a taxonomy of GP; a GP Periodic Table; mathematical formulation of GP; a GP "Pattern" book; a dictionary of pathologies of GP behavior.
- Understanding Solution Classes.
- Using tools developed in other fields to enhance our understanding and use of GP;
- How to make good use of pre- and post-processing.
- How to move beyond dumping scalars?
- Better infrastructure for visualization; probes to visualize the behavior of GP.
- More complicated fitness functions.
- Looking toward AI, aiming at "real" AI goals (but don't promise too much).
- Exploring alternative computing paradigms, beyond the microprocessor.
- How to integrate domain knowledge?
- GP as a Reinforcement Learning system.
- Scalability and Dynamics.
- Crossing the application chasm—how to make GP attractive to industry? What kind of marketing packages would be useful?

This list provides a starting point and possible directions for contributions to next year's Genetic Programming Theory and Practice Workshop. We look forward to the continued progress of theory and practice integration.

## References

O'Reilly, Una-May, Yu, Tina, Riolo, Rick L., and Worzel, Bill, editors (2004). *Genetic Programming Theory and Practice II*, volume 8 of *Genetic Programming*, Ann Arbor, MI, USA. Springer.

Riolo, Rick L. and Worzel, Bill (2003). *Genetic Programming Theory and Practice*, volume 6 of *Genetic Programming*. Kluwer, Boston, MA, USA. Series Editor - John Koza.