Challenges with Verification, Repeatability, and Meaningful Comparison in Genetic Programming: Gibson's Magic

Jason M. Daida, Derrick S. Ampy, Michael Ratanasavetavadhana, Hsiaolei Li, and Omar A. Chaudhri
The University of Michigan, Artificial Intelligence Laboratory and Space Physics Research Laboratory
2455 Hayward Avenue, Ann Arbor, Michigan 48109-2143

ABSTRACT

This paper examines some of the reporting and research practices concerning empirical work in genetic programming. We describe several common loopholes and offer three case studies—two in data modeling and one in robotics—that illustrate each. We show that by exploiting these loopholes, one can achieve performance gains of up two orders of magnitude without any substantiative changes to GP. We subsequently offer several recommendations.

1 INTRODUCTION

That researchers desire to improve genetic programming (GP) has not been in question. In the three Genetic Programming Conferences (GP'96 [Koza, et al. 1996], GP'97 [Koza, et al. 1997], GP'98 [Koza, et al. 1998]), 194 papers on genetic programming were published. Of these papers, many of them have proposed new operators, new approaches, and new paradigms. For GP'98 alone, 50 out of 67 papers published have proposed as much. That number represents 50 new and distinct enhancements that claim to improve the performance of GP— and this is just for *one* conference last year. We do not expect that trend of increasing numbers of published improvements to change anytime soon.

There is nothing intrinsically wrong about papers that propose new operators, new approaches, and new paradigms—that is part of trying to advance a field. However, given the number of new publications that claim improvements, it is inevitable that papers begin to argue for improvements that improve the prior improvements. It is also inevitable that somewhere along the way, people would *need* to compare their improvements with others'. Of course, comparisons nowadays are done in the spirit of the No Free Lunch (NFL) Theorem [Wolpert and Macready 1997]: there is no one method that is universally better than others. Because of NFL, arguments have shifted to either demonstrating application to new problem domains, or showing ways to improve on previous results. In any case, critical comparisons are inevitable, and we would argue, necessary, because the logical alternative is untenable. (The logical alternative is this: to adopt and program all new improvements whenever they are published.)

Our investigation, therefore, has focused on the GP community's reporting and publication standards. At issue is the following question: Are current reporting standards rigorous enough to allow for verification, repeatability, and meaningful comparison? In this paper, we argue that they are not, if only because one can achieve performance gains of up two orders of magnitude without any substantiative changes to GP.

1.1 GIBSON'S MAGIC

Our term—Gibson's Magic—refers to magic not of the metaphysical brand, but of the craft and art of illusion. By showing that common reporting practices in the field of GP allow for the existence of magic, we argue that we need to re-examine our reporting standards. By showing that magic is possible, we argue that our field is vulnerable to malicious disinformation and manipulation. By showing that magic is easy, we argue that researchers can commit magic even if unintended.

James Gibson, for whom we named our term, wrote in "Ecological Physics, Magic, and Reality,"

> Nevertheless, someone who knows how to manipulate and control the information available to an observer for perceiving events can make [an observer] perceive such an impossibility.… The magician does so by suppressing…information for what really happened or by preventing the observer from picking it up… [Gibson 1982, p. 219]

Dariel Fitzkee, in *Magic by Misdirection*, wrote that an audience's astonishment is

> *…caused by concealing important facts or factors or by obscuring the issues… Since* bafflement and its various shades of meaning, including mystification, mean frustration by confusion—by concealment of important factors and by making intricate—successful deception is exactly the act of doing these things plus blocking the spectator from penetrating through them to solution of the problem. [Fitzkee 1945, p. 124]

We argue, as Tufte and Swiss did for visual communication, that the practice of magic in GP is especially relevant to our reporting practices.

> To create illusions is to engage in *disinformation design*, to corrupt… information, to deceive the audience. Thus the strategies of magic suggest *what not to* do if our goal is truth-telling rather than illusion-making. [Tufte and Swiss 1997, p. 55]

1.2 ABOUT THIS PAPER

In this paper, we present three case studies. In each of these case studies, we present two different methods: Old and New. In each case study, New represents a significant improvement over Old: better than 40% improvement in one case and one

to two *orders of magnitude* improvements for the other cases. In each case, we show that these substantial improvements are not because of changes in GP, but because we have exploited loopholes in the field's current reporting practices.

Each case study is organized as follows: a problem is described; a solution is proposed; results from Old and New are given plus claims that could be made based on those results; and a discussion is offered that reveals how it was done. In each of these case studies, we specifically examine the proceedings of GP'98 [Koza, et al. 1998] and show how the reporting practices of a substantial number of those papers suffer from the weaknesses demonstrated in this paper. In Section 5, we discuss and place these case studies in the broader context of GP research practices. Section 6 lists our recommendations. Section 7 concludes our paper.

2 THE FIRST CASE STUDY

The first case study presents an argument for a new and improved version of GP called GP-SR7.2, a version of GP that has been specifically designed to work with problems concerning data modeling and symbolic regression.

2.1 (GIBSON) PROBLEM DESCRIPTION

The problem is defined as follows: fitness cases are 50 points generated from the equation $f(x) = 1 + 3x + 3x^2 + x^3$. The objective of this problem is to derive a data model that fits these points (symbolic regression). Raw fitness score is the sum of absolute error. A hit is defined as being within 0.01 in ordinate of a fitness case for a total of 50 hits. The stop criterion is when an individual in a population first scores 50 hits. Adjusted fitness is the reciprocal of the quantity one plus raw fitness score.

The function set is {+, –, ×, ÷}, which corresponds to arithmetic operators addition, subtraction, multiplication, and protected division (defined to return one if the denominator is exactly zero). A terminal set is a subset of {X, $\mathcal{R}$}, where $\mathcal{R}$ is the set of ephemeral random constants (ERCs).

The GP parameters are identical to those mentioned in Chapter 7 [Koza 1992a]: population size = 500; crossover rate = 0.9; replication rate= 0.1; population initialization with ramped half-and-half; initialization depth of 2–6 levels; and fitness-proportionate selection. Other parameter values are maximum generations = 200 and maximum tree depth = 26 (Note: these last two parameters differ from those presented in [Koza 1992a], which specifies a maximum number of generations = 51 and a maximum depth = 17. Part of the reason we extended these parameters was to mitigate against possible effects that occur when GP processes individuals at these limits.)

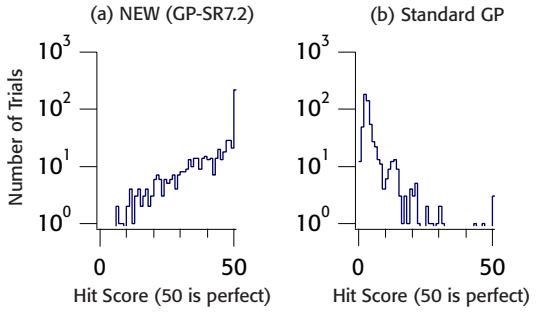

Figure 1. Hit Distribution Histograms from the First Case Study. GP-SR7.2 outperforms standard GP by a wide margin.

2.2 (GIBSON) CLAIMS AND PROCEDURE

GP-SR7.2 is a new and improved version of standard GP (which is described in [Koza 1992a]). It has been specifically formulated to work with symbolic regression and has enhanced features that allow for a manual specification of an initial population.

To demonstrate the power of our new GP-SR7.2, we compare and contrast its performance with that of standard GP on the simple problem described in Section 2.1. Furthermore, to ensure statistical legitimacy, we run 600 trials of this problem using GP-SR7.2 and another 600 trials using standard GP. Our results will show that the performance of GP-SR7.2 is vastly superior to that of standard GP.

2.3 (GIBSON) RESULTS AND CONCLUSIONS

Figure 1 shows the distribution of hits for the best-of-trial individuals. A perfect hit score is 50; a higher score is a better score. Note that the results have been plotted on a log scale.

The results show that GP-SR7.2 found perfect hit-score individuals on 219 out of 600 trials. In contrast, standard GP found just 3 perfect hit-score individuals. The performance of GP-SR7.2 over standard GP is 7200%, *which represents 1.9 orders of magnitude improvement.*

We have shown conclusively that for this instance of symbolic regression, GP-SR7.2's performance is superior to that of standard GP. Furthermore, given the very simple nature of the test problem, we believe strongly that this performance is representative of GP-SR7.2's performance on other regression problems. Unfortunately, because of the limitations of space, we leave the testing of other problems for future work.

2.4 HOW IT WAS DONE

In one sense, GP-SR7.2 was a "new" and "improved" version of GP. The actual system used was one that has been featured in [Daida, et al. 1999a]. However, for this case study, these improvements were not relevant, especially given that the nature of the improvements was to facilitate data analysis—and not to improve performance. In other words, the kernel for GP-SR7.2 was functionally equivalent to that in standard GP.

Furthermore, GP-SR7.2 and standard GP were *identical*, except for *one* parameter value. That one parameter value—ERC range—was solely responsible for the difference in performance. For GP-SR7.2, ERCs were uniformly distributed over the range [-1, 1]. For standard GP, ERCs were uniformly distributed over the range [-1000, 1000].

The loophole in standard reporting practice that we exploited is that the parameter value for ERC range does not need to be

reported. For example, of the 14 in GP'98 that involved both data modeling and ERCs, 36% did not report ERC range.

3. THE SECOND CASE STUDY

The second case study presents a revised argument for GP-SR7.2.

3.1 (GIBSON) PROBLEM DESCRIPTION

This problem description is the same as in Section 2.1, with the additional specification that ERCs that are uniformly distributed on the interval [-1, 1].

3.2 (GIBSON) CLAIMS AND PROCEDURE

The claims and procedure are the same as noted in Section 2.2.

3.3 (GIBSON) RESULTS AND CONCLUSIONS

Figure 2 shows the results from the experiment and shows the distribution of hits for the best-of-trial individuals. A perfect hit score is 50; a higher score is a better score. Note that the results have been plotted on a log scale.

As in the previous case study, the results show that GP-SR7.2 found perfect hit-score individuals on 219 out of 600 trials. In contrast, standard GP found just 18 perfect hit-score individuals. The performance of GP-SR7.2 over GP is 1117%, *which represents 1.0 orders of magnitude improvement*.

We have shown conclusively that for this instance of symbolic regression, GP-SR7.2's performance is superior to that of standard GP. Furthermore, given the very simple nature of the test problem, we believe strongly that this performance is representative of GP-SR7.2's performance on other regression problems. Unfortunately, because of the limitations of space, we leave the testing of other problems for future work.

3.4 HOW IT WAS DONE

GP-SR7.2, as mentioned in Section 2, was technically a "new" and "improved" version of GP. As before, the actual system used was one that has been featured in [Daida, et al. 1999a]. Likewise, as before, the kernel for GP-SR7.2 was functionally equivalent to that in standard GP.

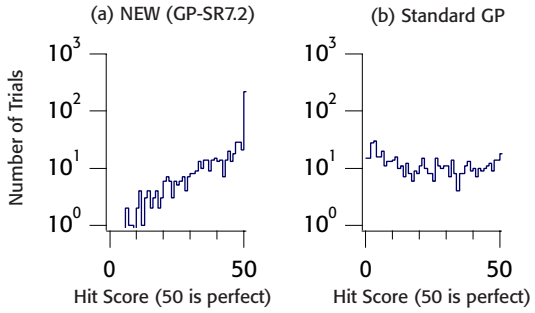

Figure 2. Hit Distribution Histograms from the Second Case Study. Again, GP-SR7.2 outperforms standard GP by a wide margin.

In particular, GP-SR7.2 and standard GP were *identical*, except for one part of the problem specification. That one specification—interval over which the fitness cases were taken—was solely responsible for the difference in performance. For GP-SR7.2, the fitness cases were taken over the interval [-1, 0). For standard GP, the fitness cases were taken over the interval [0, 1).

The loophole in standard reporting practice that we exploited is that the interval over which fitness cases are taken does not need to be reported. This shortfall was much more prevalent than the failure to note ERC ranges in GP'98. Of the 27 papers that involved data modeling and fitness-case intervals, 63% did not report the interval over which the fitness cases were taken.

4. THE THIRD CASE STUDY

The third and final case study presents an argument for a new and improved version of GP called GP-AGENT, a new version of GP that has been specifically designed to work with problems concerning agent programming and evolutionary robotics.

4.1 (GIBSON) PROBLEM DESCRIPTION

This problem is Koza's wall following robot problem [Koza 1992b], which is an adaptation of Mataric's work on a real robotics system [Mataric 1990]. The objective of this problem is to derive a program that would allow a simulated robot to follow the perimeter of an irregularly shaped room. We follow the problem specification *exactly* as is described in Koza with only two exceptions—we do not use on-the-fly redimensioning and the robot is located at a (13.8, 13.8) instead of (12, 16). (The specification covers several pages, but we recap some of those details here.) We assume that the robot is cylindrical in shape and is girded by 12 sonar sensors that allow the robot to measure the distance of each sensor to the nearest wall. Each sensor covers an area of 30°. The robot can move forward 1.0 ft, backward 1.33 ft, turn left 30° and turn right 30°. Koza and Rice's room has an irregular shape with protrusions on the south and east walls.

The terminal set consists of 12 sonar measurements `[S00, S01,..., S11]`; derived minimum of measurements `[SS]`; minimum safe distance and preferred edging distance from wall `[MSD, EDG]`; and primitive motor functions `[MF, MB, TR, TL]`.

The function set includes the If-Less-Than-Or-Equal-To macro `[IFLTE(arg1, arg2, arg3, arg4)]` (i.e., IF (arg1 ≤ arg2) then arg3, else arg4); connective function `[PROGN2(arg1, arg2)]` (i.e., eval arg1 return eval arg2).

The fitness cases are represented by Koza's irregular room with an initial starting location near the middle of the room (13.8, 13.8) and an initial facing direction of south (270°). A hit is defined when the robot touches a 2.3 ft^2 tile that exists along the walls of the room; there are 56 hits possible. Raw fitness is the number of hits in 400 time steps. Standard fitness is the total number of wall tiles minus the number of hits. The stop criterion is when the robot scores 56 out of 56 possible hits.

The GP parameters are identical to those mentioned in Chapter 13 [Koza 1992a]: population size = 1000; crossover rate = 0.9; replication rate= 0.1; population initialization with ramped half-and-half; initialization depth of 2–6 levels; fitness-proportionate with overselection; maximum generations = 57; and maximum tree depth = 17.

4.2 (GIBSON) CLAIMS AND PROCEDURE

GP-AGENT is a new and improved version of standard GP as described in [Koza 1992a]. It has been specifically formulated to work with problems concerning agent programming and evolutionary robotics. To demonstrate the power of our new GP-AGENT, we compare and contrast its performance against standard GP using the simple problem as described in the previous section. Furthermore, to ensure statistical legitimacy, we run 600 trials of this problem using GP-AGENT and another 600 trials using standard GP. Our results will show that GP-AGENT demonstrates excellent performance over that of standard GP.

4.3 (GIBSON) RESULTS AND CONCLUSIONS

The results show that GP-AGENT found 52 perfect hit-score individuals. In contrast, standard GP found just 37 perfect hit-score individuals. The performance of GP-AGENT over standard GP is 41%.

We have shown conclusively that for the wall-following robot problem, GP-AGENT's performance is superior to that of standard GP. We believe strongly that this performance is representative of GP-AGENT. Unfortunately, because of the limitations of space, we leave the testing of other problems for future work.

4.4 HOW IT WAS DONE

As before in both Sections 2 and 3, GP-AGENT was technically a "new" and "improved" version of GP. The actual system used was one that has been featured in [Daida, et al. 1999a]. However, for this case study, these improvements were not relevant, especially given that the nature of improvements was to facilitate data analysis—and not to improve performance. Again, the kernel for GP-AGENT was functionally equivalent to that in standard GP. Again, GP-AGENT and standard GP were *identical* systems with the exception that GP-AGENT used RANDU for a random number generator (see [Press, et al. 1992]) and that standard GP used the Mersenne Twister [Matsumoto and Nishimura 1998] for a random number generator. (Note: a statistical treatment is covered in Section 5.)

The loophole in standard reporting practice that we exploited is that the random number generator used does not need to be reported. Of the 65 papers in GP'98 that presented empirical results, 100% did not report the random number generator that was used to generate those results.

5. Discussion

We presented three different case studies that have illustrated Gibson's Magic. The first case study showed how one could exploit the loophole of unreported parameter values, often deemed unimportant anyway. The second case study showed how one could exploit the loophole of unreported initial conditions, which in our case amounted to what interval the fitness cases were taken. The third case study showed how one could exploit the loophole of unreported random number generators. We have shown that these three loopholes exist in 65 of the 67 papers published in GP'98. Of those papers, 16 exhibit more than one loophole.

We do not claim that any of these papers have taken advantage of any of these loopholes. We do, however, claim that the potential for error exists, particularly when another person seeks to verify, compare, or make meaningful comparisons with any of those publications. Furthermore, we have shown that the magnitude of error can be significant—up to almost two orders of magnitude in the first case study. To put that figure in perspective, we note that many papers argue that several tens of percent improvement represents significance.

◆

Suppose that the GP community moved ahead and addressed every one of those loopholes: either by amending reporting practice or by rendering a loophole inconsequential because there are better ways of doing things. (For example, [Evett and Fernandez 1998; Raidl 1998] suggest alternative methods for handling numerical constants.) Does this mean, then, that our problems have been solved?

We would argue, no, probably not. *The significance of the loopholes that we have addressed are not only that they exist, but that they point to larger problems in the field.* Because GP represents a nonlinear, bottom-up system that creates programs out of low-level structures, it is exceptionally difficult to determine in advance what is and what is not important in reporting. It is highly possible that a paper may omit what may turn out to be a crucial value, not because of malicious intent, but more likely because the authors just did not know of that value's importance.

There are a number of values and specifications that are needed to replicate a result. Koza [1992a] indicated that for (standard) GP, there are 19 control parameters. That number includes "2 major numerical parameters, 11 minor numerical parameters, and 6 qualitative variables that select among alternative ways of executing a run." [Koza 1992a, p. 114]. This number does not include any of the preparatory specifications or control parameters associated with the problem-specific code. Even for a simple symbolic regression problem like that posed in Sections 2 and 3, there are 8 specifications (i.e., objective, terminal set, function set, fitness cases, raw fitness, standardized fitness, hits, success predicate) and 5 parameters (number of fitness cases, distribution of fitness cases, interval of fitness cases, ERC range, hit criterion). The number goes up dramatically (several orders of magnitude) for more involved problems like the wall following robot, which in the early 1990s represented one of the highest-end applications for GP. For example, just in specifying the sonar-sensor maps for the wall following robot problem, an investigator needs to generate 120,000 values. Current real-world applications push the number of problems-specific parameter specifications beyond even that number.

◆

We strongly argue that because of the relative ease by which Gibson's Magic can occur in GP, we should take greater care not only in our reporting practices, but in our research practices as well. Zelkowitz and Wallace [1998] wrote a recommended article in *IEEE Computer* that took a critical look at the kinds of articles published in computer science. Their article, "Experimental Models for Validating Technology," raised questions similar to the ones offered in this paper. Just how does one determine the effectiveness of proposed theories and methods? How do we experiment? To answer their questions, Zelkowitz and Wallace surveyed and categorized several years of the following publications: *IEEE Transactions on Software Engineering*, *IEEE Software*, and *Proceedings of the International Conference on Software Engineering*. One of their categories of papers carries the label *assertive*.

An assertive paper is one in which authors propose their own technology, and then proceed to show the efficacy of that technology (empirically). While Zelkowitz and Wallace acknowledge that papers as these *do* have a place in the literature, such papers *do not* constitute the strongest measure of validation. Part of the reason is that "the experiment [in an assertive paper] is often a weak example favoring the proposed technology over alternatives. [Zelkowitz and Wallace 1998, p. 26]." Another part of the reason is because such papers are unavoidably biased.

We can see how such bias can be introduced by considering a hypothetical example. Our hypothetical example consists of the following: a real-world application of GP, say involving the design of adaptive beamforming algorithms for the control of satellite-based millimeter wavelength solid-state phased-array antennas. Each of the terms—*adaptive*, *beamforming*, *control*, *satellite-based*, *millimeter wavelength*, *solid-state phased-array*, *antennas*—implies a significant amount of graduate-level domain-specific knowledge that an average GP researcher would likely not possess. Each of these terms has an associated subfield in engineering. The application would likely require a substantial amount of mathematics and numerical methods, if the application is typical of designs in applied electromagnetics. The technologies involved in adapting GP to this application alone may be substantial and may have resulted in several new methods that could contribute to advancing GP as a field. The dilemma faced by these hypothetical authors is one faced by us: *there are only so many pages in which to describe their work*. Out of all the material they used to design their application, they *must* select only those they believe are important and essential to a GP audience. Therein lies bias that not even editors can root out.

To address bias in assertive papers, Zelkowitz and Wallace have noted categories involving independent verification, in which authors do not propose any of their own technology. Instead, such categories of papers concern analysis of others' proposed technologies (e.g., those categories include repeated experiment, synthetic environment experiments, dynamic analysis, simulation). There is value in verifying others' work, if only to root out possible biases in assertive papers. Apparently, other publications in computer science acknowledge this value—i.e., only 28% of the 152 papers that were published in 1995 and that were studied by Zelkowitz and Wallace were classified as assertive. Zelkowitz and Wallace considered that percentage to be on the high (and undesirable) side. If last year's proceedings is any reflection, GP has even further to go: about three quarters of all the GP papers published in GP'98 were assertive.

◆

Gibson's Magic provides a negative example of what one should *not* do in the reporting and research practices of the field. Indeed, that has been the primary focus of this paper. Still, we would like to explore further the consequences of our suggested alternative of investigators actively seeking out opportunities in which to verify others' technologies. We acknowledge that this alternative may not seem very palatable—after all, are not the authors whose work is being verified the ultimate reapers of fame? In the second half of this discussion, we offer the flip side of Gibson's Magic: that which makes magic possible can lead to insight, even discovery.

◆

The first two case studies featured a symbolic regression problem that involved the target function $f(x) = 1 + 3x + 3x^2 + x^3$. While somewhat typical of some of the other examples that have been cited in the literature (e.g., [Koza 1992a]), this particular polynomial has several mathematical properties that can be exploited for use in GP theory. We have called the symbolic regression problem involving $f(x)$ as the binomial-3 problem.

As it turns out, the binomial-3 problem is tunably difficult and that the tuning parameter is ERC range. (For another example of tunable problems, see [Soule, et al. 1996]. An extended list of tunable problems can be found in [Daida, et al. 1999b].) That the range of ERCs can affect problem difficulty is not new. Both [Gathercole and Ross 1996; Evett and Fernandez 1998] have noted such a phenomenon in their work. However, we have shown in [Daida, et al. 1999a; Daida, et al. 1999b] that by varying ERC range, a user can increase how GP-hard the problem is *without increasing its corresponding combinatorial search space*. Further, tuning yields definite patterns in individual size and shape.

Explaining why these patterns arise has proven to be a rich area for investigation. In an invited paper for *Advances in Genetic Programming 3*, we showed how these patterns relate to issues concerning building blocks and indicated how these patterns are not fully accounted for by GP theory [Daida, et al. 1999a]. In another paper, we showed how these patterns relate to notions of adaptive landscapes and how the metaphor and corresponding formalisms of adaptive landscapes may not necessarily apply to GP [Daida, et al. 1999b]. These are not the only investigations possible. For example, one could ask other questions like: how do these patterns change using a different operator? Is this pattern indicative of other problems? What is necessary in turning a difficult problem back into an easy one? What constrains these patterns? We note that these are only a few of the questions that arise from just the first case study.

The impetus for the second case study arose when we were algebraically simplifying the best-of-trial individuals for the

binomial-3 problem. We noted common problem-solving strategies, that instead of solving for polynomial coefficients, GP was solving for factors, i.e., $(x + 1)$. As it turns out, a common problem solving strategy for GP to employ was root finding. In retrospect, that made sense. Root-finding is a classical method for data modeling. By finding a root, GP guarantees that at least that part of the fit works. Reward (selection pressure) occurs when a root is located near fitness cases. Removal of those fitness cases that do represent roots, as the second case study showed, severely degrades GP performance. What has led to Gibson's Magic in the second case study now suggests the following general principles in using GP for data modeling: ensure that fitness cases include roots; for those models without roots, do a variable transformation such that the transformed model has roots. An investigation of these general principles would have great utility, given that at last year's conference about 40% of the GP papers published involved some sort of data model.

◆

The third case study featured the wall following robot problem, which we considered in our first "Challenges" paper [Daida, et al. 1997]. To briefly recapitulate, that paper described our efforts to replicate Koza's wall-following robot. As we mentioned earlier, the wall following robot represented one of the highest-end applications of GP in the early 1990s. Koza published several papers on the subject with co-author Rice, in addition to an extended treatment in [Koza 1992a]. What was to have been a short, two-month porting exercise for us turned out to be a long and extended lesson on verification, repeatability, and meaningful comparison. (Koza's code was written specifically for a Texas Instruments Explorer, a specialized workstation that ran its own flavor of LISP. When we had approached Koza and Rice to see if we could perhaps borrow or use the code, there was little we could use—the code ran only on a then defunct workstation. If we wanted to do the wall-following robot, the better alternative was to completely rewrite and reconstruct the code from the published record. Rice greatly helped in clarifying the published accounts.) We ended up developing our own code for the wall following robot problem (actually three separate programs) based on Koza's published accounts (e.g., [Koza 1992a]) and Rice's clarifications. One of the problems in replication and repeatability, as we subsequently identified in [Daida, et al.1997], concerned the random number generators used in GP.

The issue of random number generators turns out to be an involved investigation for which there are no clear-cut answers. That random number generators can significantly alter results is a point that we have demonstrated in [Daida, et al. 1997]. In this paper's third case study, we used two additional random number generators: RANDU and the Mersenne Twister. The infamous RANDU random number generator has been demonstrated to be not very random at all—arguably the worst of the commercial releases [Press, et al. 1992]. On the other hand, the recent Mersenne Twister [Matsumoto and Nishimura 1997; Matsumoto and Nishimura 1998] represents the other end of the spectrum and has an enormous periodicity of $2^{19,937}$ - 1, as opposed 2^{31} - 2 for some generators. The Mersenne Twister has been shown to be equidistributed randomly for 623 dimensions. In contrast, RANDU fails at 3. The first "surprise" is that a poor random number generator resulted in "better " GP performance than an excellent one. The second "surprise" is shown in Figure 3, which shows the distributions of hit scores for the wall-following robot for RANDU and the Mersenne Twister, respectively.

Unlike what we have shown in [Daida, et al. 1997], what we have shown in this paper is an instance where the distribution of hit scores are quite similar, even though intuitively one might guess otherwise. A Mann-Whitney U test of the two distributions indicate that the distributions in Figure 3 are different, but only at a confidence level of 0.51. *According to this test, the cited performance gain in Section 4 was not statistically significant*. Sampling these two distributions so that the sampling size is compatible with that presented in [Daida, et al. 1997] results in similar results of statistical significance. The Park-Miller random number generator and ran3 resulted in statistically different distributions (Mann-Whitney U test at 0.15 confidence level); while RANDU and the Mersenne Twister resulted in statistically similar distributions.

To interpret these results, we refer the reader to [Hellekalek 1997; Hellekalek 1998]. Hellekallek pointed to the crux of the problem with random number generators and quoted Compagner, saying, "Monte Carlo results are misleading when correlations hidden in the random numbers *and* in the simulation system interfere constructively [Compagner 1995] [italics ours]." The same problem also appears to hold true for genetic programming as well. That *every* random number generator has idiosyncrasies that results in *correlated* sequences is known and cannot be bypassed [Compagner 1995; L'Ecuyer 1997]. For example, RANDU fails when three or more random numbers from a sequence are taken at a time. Other random number generators may fail when parallelized [Hellekalek 1998]. The failure of a random number generator to yield random numbers, however, is consequential only if *the system in which it is used constructively (destructively) interferes with that failure*. (Not all problems exhibit this. See [Meysenburg and Foster 1999].) In our earlier paper [Daida, et al. 1997], either the subtractive generator or the Park-Miller constructively interfered with the GP system that contained the wall-

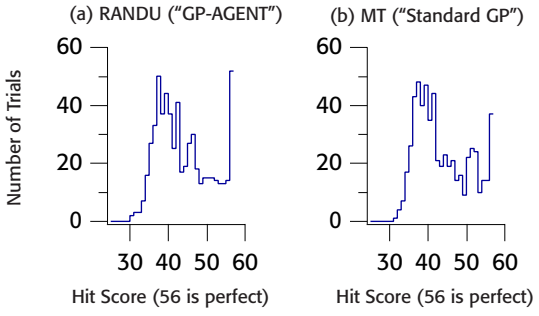

Figure 3. Hit Distribution Histograms from the Third Case Study. Although RANDU and the Mersenne Twister random number generators differ greatly in the quality of random number sequences delivered, their overall performance in GP differed only slightly, but in favor of RANDU. In other work, we have shown that this is not this always the case, using different random number generators.

following robot problem. In this paper, we show that perhaps neither RANDU nor the Mersenne Twister constructively interfered with the GP system implementing the wall-following robot (likely) or that both RANDU and the Mersenne Twister constructively interfered in approximately the same way (less likely).

◆

The evidence of these two papers suggests three items. First, that GP performance can be sensitive to the type of random number generator used. Second, that GP might be able to exploit idiosyncracies associated with a random number generator to enhance performance. Third, that analysis of just these random number generators alone does not serve as a predictor for their use in a GP implementation. These three items have two further implications. First, that empirical investigations of GP theory need to be done carefully, since patterns that may arise may be more an artifact of a random number generator than being representative of phenomena in GP. Second, that certain applications might be able to benefit from using a "loaded" random number generator. Of course, implications and insights as these have presented themselves not because the task of verification was thankless, but because the task has proven value in and of itself.

6. RECOMMENDATIONS

We offer the following recommendations based on our discussion of Gibson's Magic. (We note that the fifth recommendation is a recapitulation from our first "Challenges" paper [Daida, et al. 1997].)

1. *Report the limits and initial conditions to the problem under investigation.* For data modeling problems this includes items like the interval over which the fitness cases were taken.
2. *For those problem applications using ephemeral random constants (ERCs), identify the range and the distribution of these ERCs*. We have shown that ERCs can have a substantial effect on GP performance.
3. *State the random number generator that was used.* For some problems, which random number generator could have a bearing on performance. Furthermore, it should also be stated as to whether the random number generator was run in parallel (threaded) since parallization of random numbers generators can introduce significant correlations.
4. *Practice defensive reporting.* Assume that someone else will independently verify your work. Given that someone else may not be serving your best interests, it behooves one to exercise care in reporting. We would even suggest that defensive reporting also includes ancillary reports, data, and even some version of code to be available through one's website.
5. *Consider the use of electronic appendices.* The need to report as many specifications and parameters as is possible is usually balanced by a physical page limit. For that reason, we encourage authors to consider on-line publication of electronic appendices. Such appendices would cover in detail that which could not be discussed in a paper, but may have value in aiding others to replicate a paper's work.
6. *Recognize the value of independent verification.* There are at least two scenarios in which independent verification has a place. One involves including some level of independent verification in an assertive paper. The other (and arguably better) scenario involves crafting an entirely separate and nonassertive paper. We have discussed how Gibson's Magic should be used to evaluate for possible bias. We have also noted that there exists an significant and worthwhile value to engaging in this endeavor. We have indicated that GP as a field has a high (perhaps too high) percentage of assertive papers. We have further argued that there is a value for GP researchers to diversify their empirical work, starting first by practicing independent verification.

7. CONCLUSIONS

This paper is the second of our "Challenges" series and the third (the other was [Haynes 1998]) that examines the reporting and research practices in GP.

In this paper, we have described Gibson's Magic, and how Gibson's Magic is ripe to occur within the field because of its current reporting standards. We shown examples of how we can take advantage of Gibson's Magic and substantially improve GP performance for little or no substantiative changes to GP. We have shown performance gains of one- to two-orders of magnitude. We have demonstrated that the loopholes that have allowed for Gibson's Magic to occur are pervasive, as evidenced by the numbers of papers in last year's publication that exhibit at least one loophole. *We have noted that the significance of these loopholes is not only that they exist, but that they point to larger problems in the field.*

We have listed five recommendations for amending the reporting standards of the GP community and a sixth that encourages a change in research practice.

We have given an extended argument of why independent verification has intrinsic value to GP researchers. We have offered examples that have shown how Gibson's Magic could be used to gain insight, even discovery. We have argued that advances in GP could come not always by offering yet another technological improvement to GP, but through careful review of others' work.

◆

In the process of demonstrating Gibson's Magic, we conducted three numerical experiments. The first represented subset of experiments involving a broad numerical study on a tunably difficult problem (i.e., the binomial-3 problem). The results of this study have been featured in [Daida, et al. 1999a; Daida, et al. 1999b]. We have noted that in-depth analyses of this tunable problem has yielded insights on GP dynamics and that this problem is rich for further study.

The second indicated two general principles on applying GP for use in applications of data modeling: ensure that fitness cases include (known) roots; for those models without roots, do a variable transformation such that the transformed model has (known) roots. We have shown that the failure to supply

root information to GP can significantly degrade the performance of a GP solving a data modeling problem.

The third represents the most extensive numerical experiment on the wall following robot problem. The numerical experiment described in this paper for the wall following robot is over twice the size of previous experiments. The number of trials for this experiment (1200) has resulted in the highest resolution probability distribution of scores for this problem to date. As it has turned out, the wall following robot has served a bellwether for interactions between a random number generator and the rest of a GP system. The results from this experiment have indicated that empirical investigations of GP theory need to be done carefully, since patterns that may arise may be more an artifact of a random number generator than being representative of phenomena in GP. The results from this experiment have also indicated that certain applications might be able to benefit from using a "loaded" random number generator.

◆

For more information (other papers and code), please see our research group's site at www.sprl.umich.edu/acers.

Acknowledgments

Our work has benefitted extensively from others in our research group: R. Bertram, S. Chaudhary, J. Khoo, J. Polito 2, and S. Stanhope, for our understanding of the binomial-3 problem; G. Eickhoff, P. Litvak, and S. Yalcin, for their philosophical analysis; S. Chang, for support software; S. Ross, J.McClain, and M. Holczer, who were on the first Challenges team; K. Stubbs, for her initial work on RANDU and MT. We thank J. Koza, J. Rice, and W. Langdon for their prior discussions on this subject. We thank the reviewers for their constructive comments; those who identified themselves included J. Foster, T. Haynes, N. McPhee, and R. Keller. This research was partially supported through grants from U-M CoE, SPRL, General Electric, and UROP-OVPR. We thank J. Vesecky and S. Gregerman for their continued support. The first author thanks I. Kristo and S. Daida. In memory of D. Ternes.

References

Compagner, A. (1995). "Operational Conditions for Random-Number Generation." *Physical Review E* 52: 5634–5645.

Daida, J. M., R. B. Bertram, J. A. Polito 2 and S. A. Stanhope (1999a). "Analysis of Single-Node (Building) Blocks in Genetic Programming." In L. Spector, W. B. Langdon, U.-M. O'Reilly and P. J. Angeline (Eds.), *Advances in Genetic Programming 3*. Cambridge: The MIT Press. (In press.)

Daida, J. M., J. A. Polito, 2, S. A. Stanhope, R. R. Bertram, J. Khoo and S. Chaudhary (1999b). "What Makes a Problem GP-Hard? Analysis of a Tunably Difficulty Problem in Genetic Programming." In J. K. Koza (Eds.), *Proceedings of GECCO'99*.

Daida, J. M., S. J. Ross, J. J. McClain, D. S. Ampy and M. Holczer (1997). "Challenges with Verification, Repeatability, and Meaningful Comparisons in Genetic Programming." In J. R. Koza, K. Deb, M. Dorigo, et al (Eds.), *Genetic Programming 1997: Proceedings of the Second Annual Conference, July 13-16, 1997, Stanford University*. San Francisco: Morgan Kaufmann Publishers. pp. 64–69.

Evett, M. and T. Fernandez (1998). "Numeric Mutation Improves the Discovery of Numeric Constants in Genetic Programming." In J. R. Koza, W. Banzhaf, K. Chellapilla, et al (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22–25, 1998, University of Wisconsin, Madison*. San Francisco: Morgan Kaufmann Publishers. pp. 66–71.

Fitzkee, D. (1945). *Magic by Misdirection.* San Rafael.

Gathercole, C. and P. Ross (1996). "An Adverse Interaction Between Crossover and Restricted Tree Depth in Genetic Programming." In J. R. Koza, D. E. Goldberg, D. B. Fogel and R. L. Riolo (Eds.), *Genetic Programming 1996: Proceedings of the First Annual Conference: July 28–31, 1996, Stanford University*. Cambridge: The MIT Press. pp. 291–296.

Gibson, J. J. (1982). "Ecological Physics, Magic, and Reality." In E. Reed and R. Jones (Eds.), *Reasons for Realism: Selected Essays of James J. Gibson*. Hillsdale: L. Erlbaum.

Haynes, T. (1998). "Perturbing the Representation, Decoding, and Evaluation of Chromosomes." In J. R. Koza, W. Banzhaf, K. Chellapillaet al (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22–25, 1998, University of Wisconsin, Madison*. San Francisco: Morgan Kaufmann Publishers. pp. 122–127.

Hellekalek, P. (1997). "A Note on Pseudorandom Number Generators." *Simulation Practice and Theory* 5(6): 6–8.

Hellekalek, P. (1998). "Good Random Number Generators Are (Not So) Easy to Find." *Mathematics and Computers in Simulation* 46(5–6): 487–507.

Koza, J. R. (1992a). *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* Cambridge: The MIT Press.

Koza, J. R. (1992b). "Evolution of Subsumption Using Genetic Programming." In F. J. Varela and P. Bourgine (Eds.), *Proceedings of the First European Conference on Artificial Life: Towards a Practice of Autonomous Systems*. Cambridge: The MIT Press. pp. 110–119.

Koza, J. R., W. Banzhaf, et al., Eds. (1998). *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22–25, 1998, University of Wisconsin, Madison*. San Francisco: Morgan Kaufmann Publishers.

Koza, J. R., K. Deb, et al., Eds. (1997). *Genetic Programming 1997: Proceedings of the Second Annual Conference, July 13–16, 1997, Stanford University*. San Francisco: Morgan Kaufmann Publishers.

Koza, J. R., D. E. Goldberg, D. B. Fogel and R. L. Riolo (1996). *Genetic Programming 1996: Proceedings of the First Annual Conference: July 28–31, 1996, Stanford University.* Cambridge: The MIT Press.

L'Ecuyer, P. (1997). "Random Number Generation." In J. Banks (Eds.), *Handbook on Simulation*. New York: Wiley.

Mataric, M. J. (1990). *A Distributed Model for Mobile Robot Environment-Learning and Navigation*. Cambridge, Massachusetts Institute of Technology Artificial Intelligence Laboratory.

Matsumoto, M. and T. Nishimura (1997). *mt19937.c.* Keio, Department of Mathematics, Keio University. http://www.math.keio.ac.jp/~matumoto/emt.html.

Matsumoto, M. and T. Nishimura (1998). "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator." *ACM Transactions on Modeling and Computer Simulation* 8(1): 3–30.

Meysenburg, M. and J. A. Foster (1999). "Random Generator Quality and GP Performance." In J. K. Koza (Eds.), *Proceedings of GECCO'99.*

Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery (1992). *Numerical Recipies in C: The Art of Scientific Computing: Second Edition*. Cambridge: Cambridge University Press. pp. 274–304.

Raidl, G. R. (1998). "A Hybrid GP Approach for Numerically Robust Symbolic Regression." In J. R. Koza, W. Banzhaf, K. Chellapilla, et al. (Eds.), *Genetic Programming 1998: Proceedings of the Third Annual Conference, July 22–25, 1998, University of Wisconsin, Madison*. San Francisco: Morgan Kaufmann Publishers. pp. 323–28.

Ross, S. J., J. M. Daida, C. M. Doan, T. F. Bersano-Begey and J. J. McClain (1996). "Variations in Evolution of Subsumption Architectures Using Genetic Programming: The Wall Following Robot Revisited." In J. R. Koza, D. E. Goldberg, D. B. Fogel and R. L. Riolo (Eds.), *Genetic Programming 1996: Proceedings of the First Annual Conference: July 28–31, 1996, Stanford University*. Cambridge: The MIT Press. pp. 21–29.

Soule, T., J. A. Foster, and J. Dickinson (1996). "Using Genetic Programming to Approximate Maximum Cliques." In J. R. Koza, D. E. Goldberg, D. B. Fogel and R. L. Riolo (Eds.), *Genetic Programming 1996: Proceedings of the First Annual Conference: July 28–31, 1996, Stanford University*. Cambridge: The MIT Press. pp. 400–405.

Tufte, E. R. and J. I. Swiss (1997). "Explaining Magic: Pictorial Instructions and Disinformation Design." In E. R. Tufte (Eds.), *Visual Explanations: Images and Quantities, Evidence and Narrative*. Cheshire: Graphics Press. pp. 55–71.

Wolpert, D. H. and W. G. Macready (1997). "No Free Lunch Theorems for Optimization." *IEEE Transactions on Evolutionary Computation* 1(1): 67–82.

Zelkowitz, M. V. and D. R. Wallace (1998). "Experimental Models for Validating Technology." *IEEE Computer* 31(5): 23–31.