Vol. 43 No. SIG 10(TOM 7) IPSJ Transactions on Mathematical Modeling and Its Applications Nov. 2002

Invited Paper

Prospects for a Golden Age of Computational Innovation: How and Why Competent, Efficient Genetic Algorithms Are Changing the Future☆

David E. Goldberg†

1. Introduction

Genetic algorithms (GAs)—search procedures based on the mechanics of natural selection and genetics—are increasingly being applied to difficult problems, as this volume from the 2001 Symposium on Evolutionary Computation so strongly attests. From traditional and cutting-edge optimization in engineering and operations research to such non-traditional areas as drug design, financial prediction, data mining, and the composition of poetry and music, GAs are grabbing attention and solving problems across a broad spectrum of human endeavor. Of course, science and technology go through fads and fashions much like those of apparel, food, and toys; many practitioners are wondering whether GAs, like so many methods that have come and gone in the past, will become a permanent part of the toolkit or will fade like some computational hoola hoop du jour.

In this short essay, I argue that the former scenario is the more likely. That is, genetic algorithms—all forms of genetic and evolutionary computation (GEC)—are here to stay and will play an increasingly important role in helping people innovate in many walks of life. This may seem like a strong assertion, especially to those practitioners who have had both positive and negative experiences with genetic algorithms. But cutting-edge research suggests that the techniques that are currently in widespread use are only the tip of the iceberg. The generation of GAs that is currently in the lab promises relief from problems of scale up that some users have suffered in going from toy problems to the real McCoy. Moreover, as in so many other issues in the arena of applications, the primary determinants are often economic, not technical, and there, too, genetic algorithms have much to offer. In the remainder, I explore these issues by asking and answering the following important question: Why do real users use genetic algorithms? I will frame my answer by exploring users' motives in five categories, and this will establish the long-term utility of genetic and evolutionary computation in practical applications. I conclude by suggesting that everyday usage of these tools will usher in a golden age of computational innovation where computers will routinely help us search for better, more innovative and creative solutions to problems across the range of human endeavor.

2. Motives: Five Categories

What motivates a user to use genetic algorithms? Certainly there are as many answers to this question as there are GA users, but some generalizations can be made. Here I identify motives of five types:

(1) Motives from the buzz
(2) Motives from nature
(3) Motives from artificial systems
(4) Motives from competence
(5) Motives from economics

In the remainder of this section, we consider each of these in somewhat more detail.

3. Motives from Buzz

One of the first things that attracts new users to GAs is what I will call the "buzz". As I al-

† Department of General Engineering, University of Illinois at Urbana-Champaign, Urbana, USA

☆ Portions of this paper were originally published in the invited foreword, Goldberg, D. E. (1999), A Meditation on the Application of Genetic Algorithms, in Y. Rahmat-Samii & E. Michielssen, Eds., Electromagnetic System Design Using Evolutionary Optimization: Genetic Algorithms (pp.xxi-xxvi), New York: John Wiley. Portions of the paper are excerpted from the forthcoming book, Goldberg, D. E. (in press), The Design of Innovation, Boston, MA: Kluwer Academic Publishers.

luded earlier, GAs and evolutionary computation in general are receiving media attention, both print and electronic, and various accounts of GA discovery and invention are reverberating through popularizations of artificial life and complex systems. These accounts are often what attract new users to the field, but man does not solve problems by buzzwords alone. At some point, a problem must be posed, methods engaged, and results obtained, so motives from the buzz-while helpful in attracting new users-do little to retain them.

4. Motives from Nature

The buzz of excitement draws us to GAs, but what can keep us with them? One of the factors that certainly holds our attention is the scientific reasonableness of the endeavor. Since Darwin, we take it for granted that natural selection and natural genetics created life on this planet in all its diverse and well-adapted forms. With this understanding comes the inkling that perhaps we might be able to use nature's "search algorithm of choice" and apply it to the solution of humankind's problems.

Having this thought and making it work are two different things; yet, the inkling is important because it acts as something of an existence proof to let us know that we are on the right track even though we haven't yet engineered the ultimate genetic algorithm. Surely, people have dreamt of human flight from their first observations of birds, and for many years all attempts were doomed to failure. The knowledge that something could fly certainly played the dual role of (1) providing specific inspiration for the design details of an airplane and (2) sustaining inquiry and continued trials, especially as the failures mounted. In the same way, researchers and practitioners are inspired by nature's example and are impelled to continue even when their efforts don't turn out as they wish.

5. Motives from Artificial Systems

Nature as a source of ideas and an existence proof provides inspiration and solace, largely for the GA designer and researcher, but the practitioner's motives are rooted in the limitations of traditional optimization and operations research methods. On the one hand, there are a large number of such methods available.

When you have a linear problem with linear constraints, you can grab linear programming. When you have a stage-decomposable problem you can grab dynamic programming. When you have a nonlinear problem with nonlinear constraints, you can (sometimes) grab nonlinear programming, and so on. But the fact, that you have a list of acceptable methods for particular problem classes is itself part of the problem. Traditional methods are well tuned to a particular problem class, but when a problem comes along that violates the assumptions of such methods, solution results can be particularly disappointing. Wouldn't it be nice if artificial search and optimization procedures would work well over a broader class of problems?

Artificial genetic and evolutionary methods are a potential answer to this yearning, because the evolution of natural systems takes place via mechanisms that are in many ways invariant across species, and in so doing nature uses the same or similar search procedure almost regardless of environment. Many users turn to GAs and EC for exactly this breadth of solution quality with reasonable efficiency.

6. Motives from Competence

The promise of quality and efficiency-the promise of robustness has indeed attracted many practitioners to GAs, but for some of them, a funny thing happened on the way to their applications. At first, when working with small toy problems in their application domain, the GA works quite well, but when they turn to larger or harder problem instances they find that solution times increase, solution quality decreases, or both. The response of different users to these problems of scale up are many. Some fiddle around with operators or codings, trying different possibilities, until something works. Others abandon evolutionary computation entirely, quite frustrated with the whole affair. Others still, simply remain puzzled, and question why such ostensibly robust algorithms exhibit such poor scale-up behavior.

For years these difficulties were swept under the rug, but we now know that simple genetic and evolutionary algorithms with fixed crossover and mutation operators are fairly limited in what they can do. Mathematical analyses have been performed to support this assertion fairly convincingly, and this would seem to be a deal breaker if it weren't for companion results that show that adaptive and self-adaptive operators can overcome these difficulties quite effectively. These results have not been well integrated into practice, but as more and more practitioners become aware of them, the frustration with the problems of scale up will become decidedly less. Moreover, as these new operators take their place in everyday GA practice, users will be surprised to find that hard problems can be solved reliably and accurately in times that may grow no more quickly than a quadratic function of the number of decision variables.

7. Motives from Economics

The foregoing discussion has given a number of fairly high falutin reasons why users are motivated to use genetic algorithms, but for many practitioners the bottom line is often the bottom line. That is, practitioners are often interested in receiving economic benefits from the performance of a genetic optimization. In many cases, the economic prime movers are fairly direct. Using a genetic algorithm enables a practitioner to optimize or improve a system that is otherwise not amenable to algorithmic improvement, thereby resulting in a direct economic benefit from the use of the GA. In other circumstances, the economic benefits are somewhat less direct, but they may be critical to the choice of a GA nonetheless. We examine three such circumstances briefly:

(1) Economics of investment in method
(2) Economics of model investment
(3) Economics of GA speedup

One economic reason that users turn to GAs has to do with their investment in optimization methods. If one has limited resources and is concerned with computing improved solutions to problems with either (1) a broadly competent method such as a GA, or (2) a panoply of disparate techniques from OR or traditional optimization, the investment necessary to learn and use a single broad method should be lower than that associated with a collection of techniques. In the case of a collection of techniques, not only must many different methods be mastered, but the user must also learn when to choose which technique. These costs can add up, and other things being equal, the user may prefer to trade off the use of a perfectly tuned solver for one that does an adequate job without additional investment in knowledge of method.

Method investment costs can be significant, but for many users the lion's share of investment is tied up in modeling or simulation. Most complex optimization involves a fairly sophisticated objective function that may itself rely on finite-element models, approximations to the solutions of nonlinear equations, discrete-event simulations, or the like. Prior to using such models for optimization or design, users expend considerable time and effort inputing data, running test cases, tuning the model to agree with the real world, and then using the models for analysis. After such a large investment in modeling, no user likes to be told that in order to perform an optimization that the model must be shoehorned into a form preferred by a particular optimization method, but many optimization methods require exactly this kind of model transformation. Genetic algorithms, on the other hand, take their function evaluations as they come, thereby respecting the significant investment that users may have in analysis code, using that code without substantial modification or transformation.

This laissez faire attitude toward function evaluations comes at a cost, however. Because GAs make relatively few assumptions about the solution space, and because the interface between GA and evalution involves only the passing of function evaluation values (no derivatives or higher order information), a GA solution may require hundreds or thousands of function evaluations. As was suggested earlier, this number can be reduced through the use of competent GAs to times that may be as good as subquadratic, but nonetheless, in large problems, fairly large numbers of function evaluations will be necessary. By itself, this would be cause for some concern if there weren't corresponding ways to ways to speed up the GA itself through improved utilization of various resources, including (1) space, (2) time, (3) evaluation resources, and (4) problem specific information. These resources correspond to economies brought about through parallelization, effective continuation, function sampling and relaxation, and hybridization. Advances are begin made rapidly along all these fronts, and practitioners should soon expect to see practical means of speeding their solutions day in and day out.

8. Toward a Golden Age of Computational Innovation

Much has been made in recent times of the great strides in ubiquitous networking and the proliferation of inexpensive, powerful computers. Some go so far as to say that the industrial revolution wrought by steam power and its successors is now being overtaken by an information revolution with equally far-reaching consequences. It is hard to argue that the changes have not been dramatic, and indeed these advances have combined to reshape modern life. But it seems to me that the changes of networking are less changes in kind then they are changes in degree. Although going from telegraph to telephone to telefax to digital telecommunications represents a dramatic upgrade in speed, economy, and bandwidth delivered, it is nothing like the transformation that occurred in going from mail service to telegraph.

The "real" information revolution awaits effective computational aid for the heavy lifting of thought. Although computers now keep our books and crunch our numbers, they have not been particularly helpful in assisting our innovation and invention. This is about to change. The techniques of competent genetic algorithms—GAs that solve hard problems, quickly, reliably, and accurately—and their immediate logical extension argue for the coming of a golden age of computational innovation. In this rapidly approaching time, competent GAs and their derivatives will be used across the spectrum of human endeavor to improve all kinds of products, systems, and processes.

Already many researchers and practitioners find first-generation GAs and GEC useful in all walks of life. From engineering and the sciences to commerce to the humanities and the arts, I believe it is accurate to say that GAs are being applied more broadly and interestingly than any previous computational tool in history. Given the metaphor of evolution and genetics, perhaps we should not be surprised by this breadth, but the interesting observation is this. Widespread application of GAs has taken place despite the fact that the tools in use (first-generation GAs) are prohibitively limited in the difficulty of problems they can solve. Imagine what will happen when everyday practice adopts competent techniques. No longer will practitioners have to fiddle with codes and operators trying to get a GA that works well. Very rarely will a GA work sometimes and not others. Instead, in the near future, problems will be set up and solved, day in and day out, as a simple, unremarkable, routine matter.

And in this prospect, I believe lies the real analogy to the industrial revolution. Just as steam power initiated the release of humankind from routine physical labor, so too will competent computational innovation release humankind from routine tasks of innovation. This is an exciting prospect, because the example of the industrial revolution suggests that human beings will not be put out of business—as was feared during the industrial revolution. Instead the historical record is clear. The industrial revolution granted our species vast mechanical leverage to extend our reach beyond anything that might have been imaginable under human physical labor alone. Likewise, the coming golden age of computational innovation will release us to think at higher levels of abstraction, to assemble the combined innovative power of machine and man, to ultimately give us a kind of intellectual leverage of previously unimagined dimensions. This is the promise of competent computational innovation.

There will come a time, perhaps, when we take the power of computational innovation for granted, as we now take mechanical-electrical machines for granted. In the meanwhile, those who seize this special moment to learn about competent genetic algorithms and innovating machines-their principles of operation, their mathematical laws of behavior, and the art of their application across the range of human endeavor-will have an advantage over those who don't.

9. Conclusions

This essay started by trying to understand whether GAs are some passing fad or fancy, or whether they will become a permanent part of the problem-solving toolkit? To try to answer this, five facets or dimensions of user motivation have been examined, including motives from the buzz, from nature, from artificial systems, from competence, and from economics, and surely the real user is motivated by some combination of these factors and perhaps many others. Initially users are drawn to GAs by some combination of the first three of these reasons, but they stay for hard-headed reasons of competence, economics, or both. The essay has suggested that many of the first-generation evolutionary and genetic algorithms currently in use are incapable of solving hard problems, quickly, reliably, and accurately; in short, they don't scale up. This would be bad news if it weren't for cutting-edge research in the laboratory that shows us how to design GAs that overcome these difficulties.

Beyond the design of such competent genetic algorithms, users come and stay with GAs for a variety of good economic reasons. Certainly GAs can help directly impact the economics of design by giving us better or more cost-effective designs as the output of the optimization process. Beyond such direct impacts, users come and stay with GAs because they can reduce investment costs in methods development, because they can fully utilize existing investment in modeling and simulation, and because they can be extended to provide quality solutions more efficiently through parallelization, time utilization, relaxed function evalution, and hybridization. Together, these factors suggest that GAs will become—are becoming—a permanent part of the designer's took kit.

As this transformation takes place naturally, application by application, we shall soon find ourselves in the midst of what the essay has called a golden age of computational innovation. Just as mechanical and electromechanical machines have reduced the arduousness of physical labor, so, too, will competent genetic algorithms and their successors ease the arduousness of routine innovation and invention. Just as the industrial revolution permitted us a kind of physical leverage on our efforts, so, too, will the golden age of computational innovation give our species a kind of intellectual leverage that will multiply the consequences of our thoughts.

Acknowledgments My research was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grants F49620-94-1-0103, F49620-95-1-0338, F49620-97-1-0050, and F49620-00-0163. Research funding for this work was also provided by a grant from the National Science Foundation under grant DMI-9908252. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are my own and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Air Force Office of Scientific Research, the National Science Foundation, or the U. S. Government.

(Received October 15, 2001)
(Accepted July 17, 2002)

David E. Goldberg
Professor in the Department of General Engineering, University of Illinois at Urbana-Champaign.
遺伝的アルゴリズムの先駆的な研究で世界的に著名．有名な「Genetic Algorithms in search, Optimization & Machine Learning」の著者である．最近では「The Design of Innovation, Lessons from and for Competent Genetic Algorithms」という著書で人間の創造性の計算モデルを考えようという研究も行っている．