Applications & Science of Computational Intelligence V

Vol. 4739

PROCEEDINGS OF SPIE
SPIE—The International Society for Optical Engineering

Applications and Science of Computational Intelligence V

Kevin L. Priddy
Paul E. Keller
Peter J. Angeline
Chairs/Editors

2–3 April 2002
Orlando, USA

Sponsored and Published by
SPIE—The International Society for Optical Engineering

Volume 4739

Please use the following format to cite material from this book:

Author(s), "Title of paper," in *Applications and Science of Computational Intelligence V*, Kevin L. Priddy, Paul E. Keller, Peter J. Angeline, Editors, Proceedings of SPIE Vol. 4739, page numbers (2002).

ISSN 0277-786X
ISBN 0-8194-4489-8

Published by
SPIE—The International Society for Optical Engineering
P.O. Box 10, Bellingham, Washington 98227-0010 USA
Telephone 1 360/676-3290 (Pacific Time) • Fax 1 360/647-1445
http://www.spie.org/

Printed in the United States of America.

Conference Committee

Conference Chairs

Kevin L. Priddy, Battelle Memorial Institute (USA)
Paul E. Keller, Pacific Northwest National Laboratory (USA)
Peter J. Angeline, Natural Selection, Inc. (USA)

Program Committee

Stanley C. Ahalt, The Ohio State University (USA)
Gianfranco Basti, Pontificia Università Lateranense (Italy)
Krzysztof J. Cios, University of Colorado/Denver (USA)
David B. Fogel, Natural Selection, Inc. (USA)
Charles W. Glover, Oak Ridge National Laboratory (USA)
Mark E. Oxley, Air Force Institute of Technology (USA)
Antonio L. Perrone, Pontificia Università Lateranense (Italy)
Stephan Rudolph, Universität Stuttgart (Germany)
Eugene Santos, Jr., University of Connecticut (USA)

Session Chairs

Optimality
Peter J. Angeline, Natural Selection, Inc. (USA)

ART-Based Neural Networks
Georgios C. Anagnostopoulos, University of Central Florida (USA)

Theory
Mark E. Oxley, Air Force Institute of Technology (USA)

Applications
Emre Ertin, Battelle Memorial Institute (USA)

Introduction

This year marks the fifth anniversary of the conference on Applications and Science of Computational Intelligence. We have continually made strides in the community to improve our ability to embed intelligence in machines to make our lives better. The events of 11 September 2001 have changed the way we look at terrorism in the United States and in the world. Certainly these events have shown us that we must provide better screening techniques and better methods of identifying patterns in the vast sea of data available to us. As is often the case, investigators were able to go in after the fact and determine a pattern that pointed to a terrorist attack. We haven't had time to react to these events, but I wish to challenge all of us to think about how we can improve our ability to prevent these events from occurring in the future. Let's do that and report our research at next year's conference.

The conference this year has many papers on optimality, a panel discussion where we will consider what's next for the community, sessions on theoretical contributions, and an applications session. The poster session looks particularly impressive as well.

We are smaller this year than ever before, possibly due to the economy and the events of 11 September. We will continue to further the ideas and promise of computational intelligence. Please take time to invite your colleagues to come next year and present their ideas.

This year represents a changing of the guard once again. Dr. Paul Keller, my good friend and conference co-chair, has decided to move on to other pursuits. I will miss Paul's input and ideas for the conference. Be sure to wish him well as he moves back into optics, which he left years ago to pursue neural networks. Enjoy the new conference venue and the conference. I look forward to meeting all of you at the conference.

Kevin L. Priddy

Contents

APPLICATIONS

POSTER SESSION

Results on a fractal measure for evolutionary optimization

Peter J. Angeline
Natural Selection, Inc, 20 Hawley Street, 5th Floor West Tower, Binghamton NY*

ABSTRACT

Evolutionary optimizers employ independent Gaussian random variables as a central component for their processing, which often renders them immune to analysis. This paper investigates the applicability of the Hurst dimension, a fractal dimension, as a characterization of processing in an evolutionary optimizer. Results show that this fractal measure does highlight some interesting processing commonalities between standard and self-adaptive evolutionary optimization. A potentially worthwhile modification to evolutionary optimization is suggested based on the results.

Keywords: evolutionary optimization, self-adaptive, fractal dimension, Hurst dimension.

1. INTRODUCTION

Evolutionary optimization is a stochastic, population-based, optimization method that has enjoyed steadily increasing popularity in recent years. As with many computational techniques, the application of evolutionary optimization has advanced faster than the theoretical models of its processing. One of the difficulties in modeling evolutionary optimizations is that they are built on a stochastic foundation. Nearly every decision made in an evolutionary optimizer is based on a random distribution. In addition, these optimizers embody a competitive process that makes many of the stochastic decisions relative to the current state of the optimization. In addition, when the standard algorithms are augmented, such as in self-adaptive evolutionary optimization, the interaction of the various stochastic processes changes significantly. Not surprisingly, models and metrics for the processing in these algorithms are often tied to a specific algorithm applied to a specific problem class.

The goal of the experiments described below is to investigate the applicability of a specific fractal dimension, termed the Hurst dimension, as a measure for both standard and self-adaptive evolutionary optimizers. The fractal measure is not applied directly to the optimizer but to the sequence of random values generated by the stochastic process during optimization that give rise to the optimal solution. The results show that the metric produces similar values for both styles of evolutionary optimization within a problem and that some commonality between problems can be identified.

This paper begins with an in-depth description of evolutionary optimization, and the Hurst dimension. It then describes the method used to measure the fractal dimension of an evolutionary optimizer running on a problem. This is followed with a description of the test cases used and other experimental details. The paper concludes with a report and discussion of the experimental results and some suggestions for future work.

2. BACKGROUND

2.1 Evolutionary optimization

Evolutionary computations ([8], [5], [10]) are stochastic methods for population-based search and optimization with the following form:

$$P' = \mu(s(f(P))) \tag{1}$$

* The author can be reached at angeline@natural-selection.com.

Applications and Science of Computational Intelligence V, Kevin L. Priddy, Paul E. Keller,
Peter J. Angeline, Editors, Proceedings of SPIE Vol. 4739 (2002) © 2002 SPIE · 0277-786X/02/$15.00

where μ is a mutation function that randomly varies a subset of the individuals in the population, and s is a selection function that removes poorly performing individuals and replaces them with copies of other population members, called *parents*.

[1] categorizes evolutionary optimizers using two distinct dimensions. The first dimension asks at what level the adaptive parameters are associated with the evolutionary computation. *Population-level* adaptive parameters are adaptive parameters that affect the entire population, for instance the rate of crossover or mutation performed on the evolving population. *Individual-level adaptive* parameters are associated with each individual in the population and determine which component in the individual is to be manipulated. *Component-level* adaptive parameters are also associated with individuals but determine instead the severity of manipulating a particular component in the individual.

The second dimension used to categorize adaptive evolutionary computations in [1] is the method's adaptive parameter *update rule*. Update rules are applied to the adaptive parameters in order to modify their values over time to more accurately reflect the optimal parameter settings given the current environment. There are essentially two forms of update rules currently in use. *Absolute update rules* are statistical measures or similarly induced heuristics that are determined prior to a run and dictate in an absolute manner how the adaptive parameters change over time. *Empirical update rules* are a distinct type of update rule that uses the dynamics of the evolutionary process to determine the appropriate values for the adaptive parameters. Empirical update rules are typically more reactive to the idiosyncrasies of the particular landscape being traversed. Evolutionary computations that use empirical update rules are usually called *self-adaptive*. Implementation of the update functions usually reside in the mutation operation.

Mutation operations in evolutionary computations take many forms. The exact form used is typically selected with an understanding of the representation to be manipulated. In evolutionary optimization, where the individuals are fixed length real-valued vectors, mutation can be performed as follows:

$$x_{ij}' = x_{ij} + \alpha N(0, 1) \tag{2}$$

where x_{ij} is the jth position of the vector stored by the ith individual, N(0,1) is a Gaussian random variable with mean 0 and standard deviation of 1, and α is a constant reflecting the standard deviation applied to the noise which acts as a mutation step size. This formulation represents the most basic form of evolutionary optimization and is termed *standard evolutionary optimization* in this paper.

In general, the majority of evolutionary optimizers employed incorporate self-adaptation chiefly to modulate the severity of the mutations. There are a number of methods for this type of self-adaptive evolutionary optimization, differentiated by their choices for the update rules (e.g. [6], [11], [10], and [13]). The mutation used in the current study is:

$$x_{ij}' = x_{ij} + \Psi(\sigma_{ij}) \tag{3}$$

$$\sigma_{ij}' = \Upsilon(\sigma_{ij}) \tag{4}$$

where x_{ij} is defined as above, σ_{ij} are the *strategy parameters* for individual i, $\psi(\sigma_{ij})$ is the update function applied to component j of individual i, and $\Upsilon(\sigma_{ij})$ is the update function applied to strategy parameter j of individual i. In self-adaptive methods, the strategy parameters determine the severity of the mutation applied to the individual and are co-evolved within the individual in order to adapt their values to what is best suited to the search space region the individual currently occupies. Research in [10], [11], and [2] has shown that an exceptional choice for these functions is as follows:

$$\psi(\sigma_{ij}) = \sigma_{ij} N_{ij}(0, 1) \tag{5}$$

$$\Upsilon(\sigma_{ij}) = \sigma_{ij} \exp(\tau' N_i(0, 1) + \tau N_{ij}(0, 1)) \tag{6}$$

$$\tau = \sqrt{2\sqrt{n}}^{-1} \tag{7}$$

$$\tau' = \sqrt{2n}^{-1} \tag{8}$$

where $N_i(0,1)$ is a gaussian random variable generated once for individual i, $N_{ij}(0,1)$ is a gaussian random variable generated once for each component in individual i, with the τ and τ'parameters typically set to the values shown in Equations 7 and 8. Equation 6 implements a lognormal random update function for the strategy parameters which has been shown to be superior on average to the alternate Gaussian update function for strategy parameters ([2], [12]).

2.2 Fractional Gaussian noise and the Hurst dimension

Stationary Gaussian distributions, also called normal distributions, are often assumed to be ubiquitous in science and nature. This is reflected prominently in signal processing where Gaussian noise is often a default assumption when trying to detect a signal in an incident data stream. The argument is that any noise mixed with a signal is due to natural phenomena and hence will be uncorrelated and follow a Gaussian distribution. A data stream is correlated if the values of the data stream at different times are not independent. However, in many natural and man-made processes, the data stream is not independent but often displays some form of correlation.

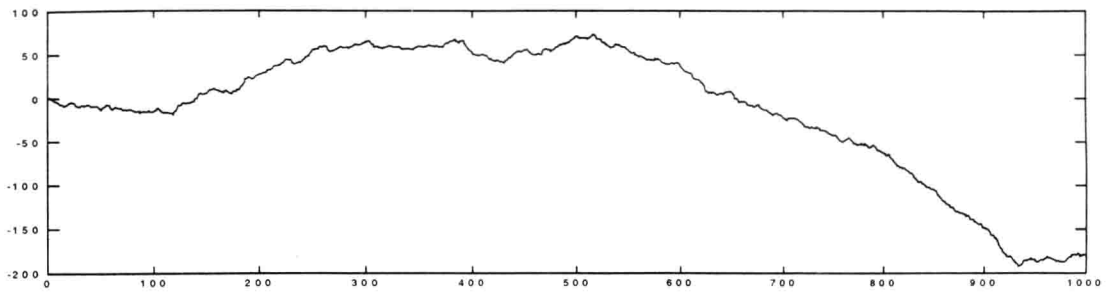
One form of correlation in a time-series produces a Gaussian distribution over a long sample period while the individual values are correlated to values that occur close to it in the series. Hurst [7], noticed such a phenomena in his studies of the rainfall in the Nile river basin and developed a method, called *rescaled range analysis*, to measure the degree of local correlation in a signal. In fact, what Hurst had discovered was the first fractal dimension and a method for measuring it in a time-series. This form of Gaussian "noise" has since become known as *fractional Gaussian noise* (fGn). And, just as the integral of Gaussian noise is Brownian motion, the integral of fractional Gaussian noise is *fractional Brownian motion* (fBm). Later, this metric was named the Hurst dimension by Mandelbrot [9] and is denoted with the symbol H in honor of Hurst.

The value H for a given time-series is defined as a value between 0.0 and 1.0 reflecting the degree of autocorrelation present in the time-series. An H value of 0.5 denotes that the time-series is completely uncorrelated and hence that all values are independent. Standard Gaussian noise can thus be defined as fractional Gaussian noise with H=0.5. A value of H greater than 0.5 denotes that the time-series is positively correlated. This means that surrounding values in the series are likely to be near to the current value. The degree of positive correlation is reflected in how close the H value is to 1.0. Similarly, an H value less than 0.5 denotes that the series is negatively correlated, which means that surrounding values are likely to be far from the current value. The degree of negative correlation is reflected in the proximity of the value to 0.0 with 0.0 denoting complete negative correlation.

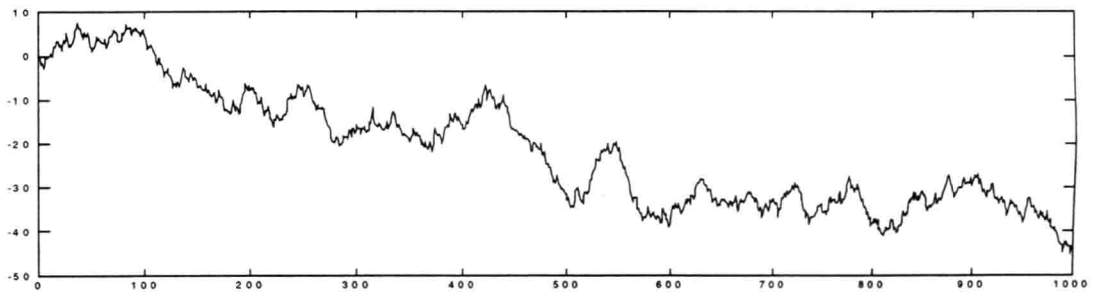
As an illustration of the effects of H on a time-series, Figure 1 shows examples of fBm with H values of 0.75, 0.5, and 0.25. Higher values for H are reflected visually as smoother graphs while lower values for H create time-series that appears more rugged. It is interesting to note that Hurst [7] found H= 0.7 was very common in the natural phenomena he studied, which suggests that those natural phenomena tended to be locally correlated.

Rescaled range analysis, Hurst's [7] method for estimating H, is simple but not very accurate. A more accurate alternative for estimating H is the *scaled windowed variance* (SWV) method [3]. Figure 2 illustrates the steps used when calculating H using this technique, which are as follows:

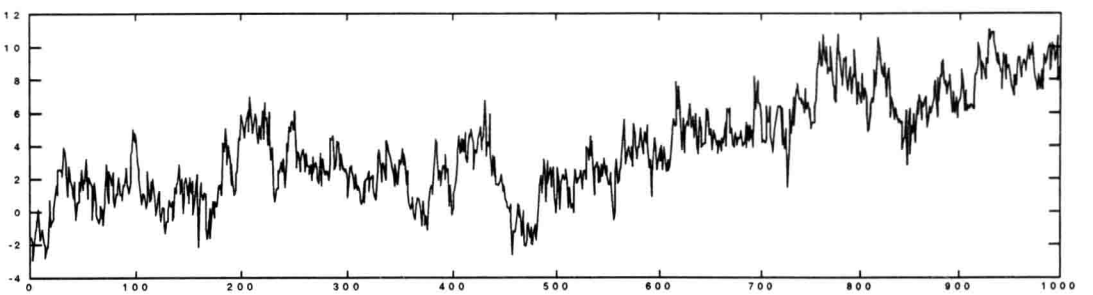
1. Divide the time-series into separate consecutive windows of the same size for all integer window sizes.
2. Find the average standard deviation for each window size over the set of windows.
3. Plot the log of the average standard deviation to the log window size.
4. Compute the least-squares linear fit for the plot and return the slope as H.



(a)



(b)



(c)

Figure 1: Examples of 1000 consecutive steps of fractional Brownian motion, the integral of fractional Gaussian noise, for three values of H. (a) fBm with H=0.75; (b) fBm with H=0.5 which corresponds to standard Brownian motion; (c) fBm with H=0.25. Note that as the value for H decreases the graph of the fBm appears more rough. Note also that the range traversed by the fBm decreases as H decreases. These series were produced using the off-line technique for creating fGn described in [4].

The work in [3] has shown this procedure and other similar methods produce much more accurate estimates of H. SWV will be used to measure the H dimension of all generated time-series in the experiments below.

5. EXPERIMENTAL METHOD

When an evolutionary optimizer runs, there must be a continuous progression of individuals starting from some individual in the initial randomly selected population, and the best individual of the final generation. This set of individuals represents the succession and progression of solutions from those values assigned randomly in the first individual to those

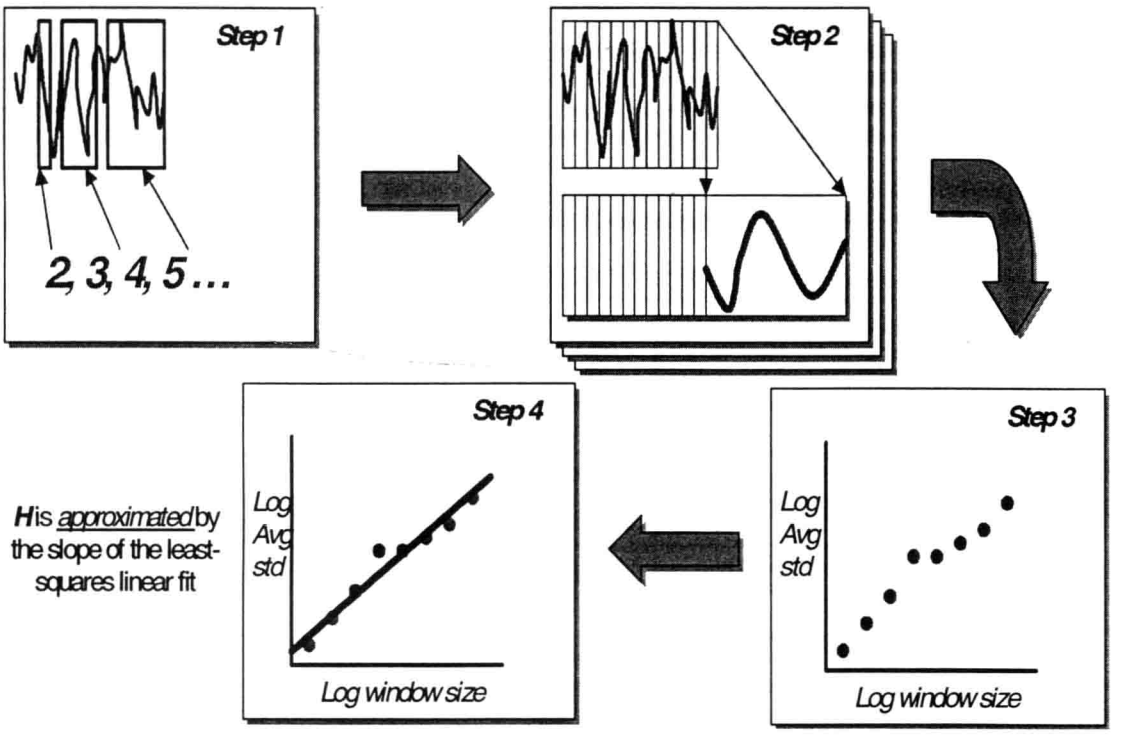


Figure 2: The steps involved in approximating H for a given time-series. In Step 1, a set of window sizes is determined. Usually these are selected to be integral divisors of the sequence length. In Step 2, the standard deviation of all windows of the time-series are calculated and averaged. Step 3 involves plotting the log of the window size against the log of the average standard deviation found for that window size. Finally, H is approximated in Step 4 by the slope of the least squares linear fit of the plot generated in Step 3.

present in the last. Each step in this succession was generated from a independent Gaussian random variable, here either using equation 2 or 5 above. The question this experiment addresses is if the series of independent values pulled from the Gaussian random variable that created the progression of individuals from the first generation to the last has properties that are distinct from its generator.

In each experiment below, the set of random values used to produce all offspring from parents during the evolutionary optimization process is saved for analysis after the run. At the conclusion of a run, the progression of individuals that gave rise to the best individual in the last generation is reconstructed. A set of time-series is then created, one for each dimension of the objective function, that contain the progression of random values applied to that dimension from the first individual to the last. For instance, if a 30 dimensional function is being optimized then 30 time-series, one for each dimension in the evolving vector, are constructed. SWV analysis is then applied to each individual time-series to estimate its H value.

Given that all of the random values used in the evolutionary optimization are generated using independent pulls from a Gaussian random variable, there is not a priori reason to assume that any subset of values taken from the evolutionary process will be anything but independent. As a control, a similar process to the above is performed except that the set of individuals used to create the time-series is created from the random values that were applied to any parent to create the best of generation individual for each generation. Other than this difference, the method was identical.

Six numeric functions well studied in the evolutionary optimization literature were used as test functions in the experiments. They are:

$$\text{Sphere}(x) = \sum_{1}^{n} x_i^2 \tag{9}$$

$$\text{Rastrigin}(x) = \sum_{1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) \tag{10}$$

$$\text{Griewank}(x) = \frac{1}{4000}\sum_{1}^{n} x_i^2 - \prod_{1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{11}$$

$$\text{Rosenbrock}(x) = \sum_{1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{12}$$

$$\text{Schwefel2.21}(x) = \max_i(|x_i|,\ 1 \le i \le n) \tag{13}$$

$$\text{Schwefel2.22}(x) = \sum_{1}^{n} |x_i| + \prod_{1}^{n} |x_i| \tag{14}$$

all of which are taken from [13] who collected them from other sources. The optimum for each of the above functions is reached when all dimensions in the solution vector are set to 0.0. Following [13], the initialization limits for the initial population for each function were set to the values shown in Table 1.

Table 1: Initialization ranges for each of the test functions

Function	*Range*
Sphere	$[-100, 100]^n$
Rastrigin	$[-5.12, 5.12]^n$
Greiwank	$[-600, 600]^n$
Rosenbrock	$[-30, 30]^n$
Schwefel2.21	$[-100, 100^n$
Schwefel2.22	$[-10, 10]^n$

A total of ten trials were performed for each of the above functions using both standard evolutionary optimization and self-adaptive evolutionary optimization. For each trial, the set of time-series of random values generated to create the lineage for the best overall individual was reconstructed for each dimension as well as the set of time-series constructed for the best of generation individuals. The H value for each dimension in each trial of both was estimated and the average of all H values for all dimensions over all trials was calculated. In each trial, the population size was set to 50 individuals with 50% of the individuals replaced by offspring after each generation. Each trial on each function was run for a total of 2000 generations.

6. RESULTS

For all trials and all functions, without exception, the mean H found for the time-series generated from the best of generation individuals was 0.5 with extremely small standard deviations. This was consistent across both the set of problems and the two evolutionary optimization techniques investigated. This suggests that these time-series were very nearly if not completely independent, showing very little or no autocorrelation in the reconstructed time-series.

The results for the experiments involving the lineage of individuals leading to the overall best individual are presented in Table 2. The table shows the average and standard deviation of H estimated for the constructed sequences for both self-

Table 2: Results for the experiments described above

Function	*Average H without self-adaptation*	*Standard Deviation of H without self-adaptation*	*Average H with self-adaptation*	*Standard Deviation of H with self-adaptation*
Sphere	0.7207	0.1196	0.6902	0.1261
Rastrigin	0.6386	0.1108	0.6354	0.0963
Greiwank	0.7059	0.1328	0.710	0.1272
Rosenbrock	0.6421	0.1003	0.6832	0.1121
Schwefel 2.21	0.6746	0.0995	0.6682	0.1022
Schwefel 2.22	0.5781	0.1245	0.5744	0.1167

adaptive and non-self-adaptive evolutionary optimization for each of the six test functions investigated. The moderately large standard deviations show that there was a fair amount of variation across the 30 dimensions of the individuals. Note that the average H for both evolutionary optimization techniques are similar for each problem. Also note that all average H values shown in the table are greater than 0.5, suggesting that the time-series showed a tendency toward persistence, meaning that values relatively close in the series tended to be similar in value.

7. DISCUSSION

First, the fact that all the results shown in Table 2 indicate the random values used to create the lineage that lead to the best overall individual were positively correlated is interesting. This is in direct opposition of the control experiment using the best of generation individuals, which showed no tendency for persistence. This suggests that successful lineages in both of the evolutionary optimizers investigated tend to generated by sequences of random numbers that follow a consistent direction for each individual dimension. In one sense, this is not surprising, since the shortest path to any optima is a straight line. In addition, the smoother objective functions, such as the sphere and Griewank functions, tended to produce more time-series with larger H values, i.e. with more persistence.

One surprising aspect of the results above is that the average H values were consistent between the two different evolutionary optimizers for all functions. Given the ability of the self-adaptive process to adapt its step size dynamically and the fact that these evolutionary optimizers find solutions more quickly, it might be expected that their H values would be larger than for standard evolutionary optimization. The results here do not support that expectation.

The above results suggest an interesting and potentially worthwhile enhancement for evolutionary optimizers. Since the best lineages displayed persistence in their pattern of random values, replacing the independent Gaussian random variable in equations 2 and 5 above with persistent fractional Gaussian noise, i.e. fGn with $H>0.5$, could lead to faster optimization. In essence, some sort of momentum term involving successful mutations might be incorporated as a guide for the mutation process. One possible implementation of this idea is to use a fGn generator in place of the appropriate indepen-

dent Gaussian random variables. The difficulty with this approach is that no on-line method for generating fGn exists. Attempts by the author to employ off-line fGn generators for this task, such as that described in [4], have yet to produce fGn sequences with consistent H values. This remains an issue for further study.

8. CONCLUSIONS

This paper has described research showing that the random values that form the lineage of individuals the result in the best overall individual in two forms of evolutionary optimization consistently display positive correlation to local values in the series. This effect is consistent across the two techniques investigated over the six objective functions. In addition, the results above show that the magnitude of the effect is consistent within each test function across the two evolutionary optimizers. The results suggest that the successful lineages in evolutionary optimization are biased towards locally consistent mutations, mutations that carry some momentum when successful. The discussion above suggests that a modification to the mutation process that biases mutations to be more consistent may provide a worthwhile modification to evolutionary optimization.

ACKNOWLEDGEMENTS

Thanks to M. Nowatarski and other attendees of the Symposium on Chaotic Systems who viewed and commented on a preliminary version of this work. Thanks also to D. Percival who answered several questions via email regarding the generation of fGn and alternative methods for estimating the Hurst dimension.

REFERENCES

1. Angeline, P. J., "Adaptive and self-adaptive evolutionary computations" In *Computational Intelligence: A Dynamic System Perspective*, M. Palaniswami, Y. Attikiouzel, R. Marks, D. Fogel and T. Fukuda (eds.), Piscataway, NJ: IEEE Press, pp. 152-163, 1995.
2. Bäck, T. and Schwefel, H.-P. "An overview of evolutionary algorithms for parameter optimization," *Evolutionary Computation*, **1** (1), pp. 1-24, 1993.
3. Cannon, M., Percival, D. Caccia, G. Raymond, J. Bassingthwaighte, "Evaluating scaled windowed variance methods for estimating the Hurst coefficient of a time series," *Physica A*, **241**, pp. 606-626, 1997.
4. Davies, R. and Harte, D., "Tests for Hurst effect", *Biometrica*, **74**, *1*, pp. 95-101, 1987.
5. Fogel, D.B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*, Piscataway, NJ: IEEE Press, 1995.
6. Fogel, D.B., Fogel, L.J. and Atmar, J.W., "Meta-evolutionary programming." *Proc. of the 25th Asilomar Conference on Signals, Systems and Computers*, R.R. Chen (ed.), San Jose, CA: Maple Press, pp. 540-545, 1991.
7. Hurst, H., "Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers*, 116, pp. 770- 808, 1951.
8. Fogel, L.J., Owens, A.J. and Walsh, M.J., *Artificial Intelligence through Simulated Evolution*, New York: John Wiley, 1966.
9. Mandelbrot, B. and Van Ness, J., "Fractional Brownian motions, fractional Gaussian noises and applications," *SIAM Review*, **10**, 4, 1968.
10. Schwefel, H.-P. (1981). *Numerical Optimization of Computer Models*. Chichester, U.K: John Wiley.
11. Saravanan, N. and Fogel, D. B., "Learning strategy parameters in evolutionary programming: An empirical study," In *The Proceedings of the Third Annual Conference on Evolutionary Programming*. A. Sebald and L. Fogel (eds.). New Jersey: World Scientific. pp. 269-280, 1994.
12. Saravanan, N., Fogel, D. B. and Nelson, K. M., "A comparison of methods for self-adaptation in evolutionary algorithms," *BioSystems*, **36**, pp. 157-166, 1995.
13. X. Yao and Y. Lin, "Fast evolutionary programming," in *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*, L. Fogel, P. Angeline, and T. Bäck (eds.), Cambridge, MA: MIT Press, pp. 451-460, 1996.

Genetic routing algorithms to optimize availability in broadband wireless networks with load balancing

William S. Hortos
Florida Institute of Technology, Orlando Graduate Center, 3165 McCrory Place, Suite 161, Orlando, FL 32803

ABSTRACT

Packet-switched networks using the Internet Protocol (IP) provide multimedia services through broadband wireless access to mobile and fixed subscribers from an IP core network via bi-directional paths consisting of a hierarchy of high-speed routers, switches, and servers. Packets are aggregated at the nodes that form the ordered links of end-to-end paths between subscriber and gateway. Network resources are allocated at nodes to meet quality of service (QoS) requirements of new and existing calls. If sufficient resources are not available to satisfy a call's QoS, the call is blocked or dropped, reducing network "uptime" or availability. Packet flows are shared among redundant devices, clustered at nodes, to reduce blocking and dropping and speed failure recovery. A two-stage genetic algorithm (GA) is proposed to assign resources to feasible paths to provide calls the best possible resource utilization, availability, and QoS levels, while balancing traffic among devices at nodes. The GA operates on a population of integer-valued vectors of call ID, QoS requirements, and end-to-end paths encoded as node-device pairs. Selection, crossover, and mutation are defined for the GA. At call arrivals and departures, the GA limits the number of candidate paths based on their fitness to provide QoS, path availability, resource utilization, and load balance. Simulation results are discussed for different scenarios.

Keywords: Broadband wireless access; packet-switched networks; Internet routing protocols; multimedia services; genetic algorithms; optimal resource allocation; quality-of-service (QoS) routing; load balancing; network availability

1. INTRODUCTION

Providers of multimedia services are challenged to transition from circuit-switched ATM and frame relay which are already deployed in large-scale networks to build large-scale Internet Protocol (IP)-based networks with the capabilities to provide aspects of quality of service (QoS)/class of service (CoS) and facilitate the use of virtual private networks (VPNs). Added to these challenges are market needs to extend these services to mobile subscribers through broad wireless access technologies based on third- and fourth-generation industry standards. The Internet Engineering Task Force (IETF) is defining new IP concepts in response to numerous interrelated problems to establish large-scale, hybrid-access IP networks. These problems include scaling IP networks to meet the growing demands of Internet traffic, enabling differentiated levels of IP-based services to provisioned, merging traffic types of varying QoS requirements onto a single IP network, and improving operational efficiency in a dynamic environment.[1, 2]

The deployment of new protocols, such as, Multiprotocol Label Switching (MPLS), combining layer 2 (data link layer) switching with layer 3 (network layer) routing, will satisfy the requirements of these service providers.[3] Label-switching technology is a result of the desire to combine the benefits of switching technologies in the core of the network with the benefits of IP routing technologies at the edge of the network. A hybrid network based on both of these technologies as well as broadband wireless access technology and protocols creates a larger problem best described as "how to make IP, ATM and broadband wireless inter-operate." Label switching seeks to combine the best attributes of layer 2 switching, as embodied in ATM and frame relay, with the best attributes of the layer 3 routing embodied in the IP domain. MPLS, as the standards-based approach to label switching, identifies and marks IP packets with labels and forwarding them to modified switch or router, which then uses the labels to switch the packets through the network. The labels are created and assigned to IP packets based upon the information gathered from existing IP routing protocols.

The resulting network architecture for delivering IP traffic places switching at the core of the network, while IP routing continues to dominate the edge. The need to integrate the two different technologies has given rise to the use of overlay networks where the access technology (IP) has been overlaid on the core technology (ATM or frame relay), and,

Applications and Science of Computational Intelligence V, Kevin L. Priddy, Paul E. Keller, Peter J. Angeline, Editors, Proceedings of SPIE Vol. 4739 (2002) © 2002 SPIE · 0277-786X/02/$15.00