Using Time Efficiently: Genetic-Evolutionary Algorithms and the Continuation Problem

David E. Goldberg
Illinois Genetic Algorithms Laboratory
Department of General Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801 USA

Abstract

This paper develops a macro-level theory of efficient time utilization for genetic and evolutionary algorithms. Building on population sizing results that estimate the critical relationship between solution quality and time, the paper considers the tradeoff between large populations that converge in a single convergence epoch and smaller populations with multiple epochs. Two models suggest a link between the salience structure of a problem and the appropriate population-time configuration for best efficiency.

1 INTRODUCTION

Great strides have been made in our understanding and design of a variety of diversity prolonging or rejuvenating operators in genetic-evolutionary algorithms (GEAs). Such diverse mechanisms as adaptive and self-adaptive mutation operators (Bäck & Schwefel, 1995), dominance and diploidy (Goldberg & Smith, 1987), and even linkage learning (Harik & Goldberg, 1997; Harik, 1997) have been used to prolong or rejuvenate diversity in the face of continued selection and the vagaries of genetic drift. Despite this apparent progress at the micro- or operator-design level of detail, less has been said regarding the need for such mechanisms at the macro-level of solution quality and speed.

This paper takes a number of steps in the direction of remedying this situation. Specifically, the paper derives two idealized macro-models of the interaction between run duration, population size, and solution quality and then uses those models to investigate the most efficient configuration of a GEA in time. Although idealized, the modeling does consider real-life difficulties such as the cost of *rework* when diversity rejuvenating operators mistakenly diversify genes that have converged correctly. Together, the modeling and its application help explain the important role of diversity prolongation and rejuvenation operators in *continuing* the search in time.

There can be a variety of reasons why runs must be continued. Perhaps most commonly, continuation is necessary because salient building blocks attract the greatest attention early on, and badly scaled building blocks then must be preserved or resurrected if they are to be properly searched subsequently in the run. In other kinds of problems, certain allele combinations cannot be properly determined until others have converged to their proper value (van Nimwegen & Crutchfield, in press). In either case, there is an essential dimension of serial search that must be followed, and this raises the need for operators that continue the run in time.

At one level, this *continuation hypothesis* should help us understand how to configure GEAs to solve problems to a given level of accuracy most quickly. At another level, I believe it will help us redraw the battle lines between proponents of recombination on the one hand and mutation on the other. Viewed from the perspective of continuation, the issue is not recombination versus mutation, but instead how to combine the two to best benefit the overall GEA.

We start by examining the fundamental tradeoff between population size and and run duration. We continue by reviewing the building-block theory necessary for building the continuation models. We then construct two continuation models, one appropriate for problems with building blocks of uniform salience, and one appropriate for building blocks of decreasing salience. We then look at some supporting evidence in the literature, and conclude with a number of continuations and extensions of the work.

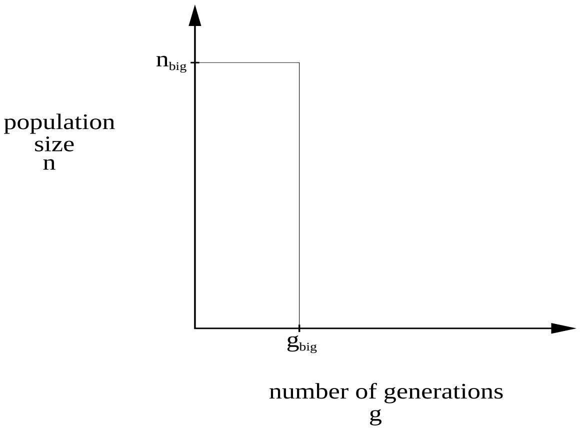

Figure 1: A run with a large population size n_{big} terminates in a single epoch with an accurate solution
.

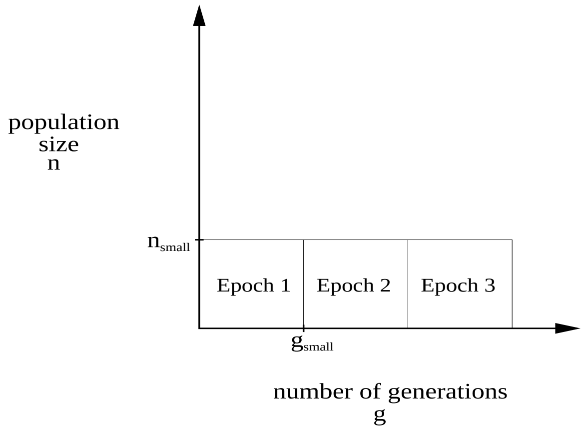

Figure 2: A run with a small population size may run a larger number of epochs and consume the same amount of time as the large population situation depicted in figure 1
.

2 THE FUNDAMENTAL TRADEOFF

This section explores the fundamental tradeoff that often exists in GEA runs. Consider the situation depicted in figure 1. There, a population has been sized large enough ($n = n_{big}$) such that the desired solution quality (solution accuracy and reliability) is obtained. More will be said on how to do such sizing in a moment, but for now suffice it to say that large, complex problems, with badly scaled building blocks need larger populations than smaller, simpler, more uniformly scaled problems. At the end of the run of figure 1, a total number of function evaluations

$$T = n_{big} \cdot g_{big} \tag{1}$$

will have been performed. Here, T is the total number of function evaluations, and g_{big} is the number of generations to completion of the run.

Contrast this situation to that depicted in figure 2, where a smaller population has been chosen. Here, we start by assuming that the GEA is run to substantial convergence under the action of selection and a mixing operator such as crossover. At the end of this first convergence *epoch*, the use of a smaller population compared to that of figure 1 leaves us with a poorer quality solution. But the comparison is not fair, because the use of a smaller population means that we have not expended the same number of function evaluations as in the n_{big} case. This opens the door to the possibility of applying a diversity rejuvenating or what I shall call a *continuation* operator to preserve or provide appropriate diversity to permit the search to go on. Here, we imagine this procedure continuing epoch by epoch, until such time as the number of function evaluations is the same as in figure 1. This condition will occur when

$$T = e \cdot n_{small} \cdot g_{small} \tag{2}$$

Here e is the number of epochs of population size n_{small} and length g_{small} used in the run.

Having constructed two situations, each with a common number of function evaluations (and thus requiring approximately the same amount of clock time on a serial computer), we might then ask, which of the two situations is superior in generating a solution with higher quality? The answer is "it depends," and a large portion of the remainder of the paper is devoted to giving a rational, quantitative answer to this important question. It should be pointed out that this tradeoff between large and small population n values has been understood since the early days of rational population sizing (Goldberg, Deb, & Clark, 1992; Goldberg & Rudnick, 1991) and indeed a number of operators that qualify as effective continuation operators have been developed, although that term has not been used before—except in my talks over the last two years—nor has their function been viewed through this lens.

We will take up the challenge of modeling continuation in time in a moment, but first we consider the characteristics of the perfect operator for continuation.

2.1 IDEALIZED CONTINUATION OPERATOR

Suppose we could construct the perfect operator for continuing a run—an *idealized continuation operator* (ICO). How should it behave? Again assuming that selection plus mixing operates in isolation and to substantial convergence, we might imagine that an ICO would then do three things:

1. Leave correctly converged alleles (those that agree with an optimal or target solution) alone.
2. Substantially perturb improperly converged alleles.
3. Achieve conditions (1) and (2) without additional cost to the solution process.

Of course, an ICO is idealized precisely because it is not possible to do these three things exactly, but we can measure the quality of a continuation operator by how closely it achieves the ideal. Moreover, we can start our analyses of continuation from the viewpoint of an ICO and then modify that analysis to reflect some cost associated with trying to do items (1) and (2).

2.2 CONTINUATION ERRORS AND THEIR COST

Thinking about the perfect continuation operator immediately begs us to think about the ways in which a real continuation operator (RCO) might go wrong and apparently there are at least two:

1. Type I error: Perturb a good allele.
2. Type II error: Leave a bad allele alone.

GEAs with background mutation rates are unlikely to make type II errors permanently, and here we focus on the type I error. Adaptive mutation operators and expression-abeyance operators can learn to mutate the right alleles at appropriate rates, but there is still the issue of the length of time that it takes to stop perturbing good stuff. Moreover, if background mutation is employed as assumed above, there will always be some non-zero (albeit small) probability of revisiting properly converged alleles.

Both of these effects lead us to consider the cost of *rework*, whereby alleles that have been properly decided must be revisited and decided again. Such costs are especially pernicious in cases where a *salient* building block or allele is perturbed. More will be said about this in a moment, but a building block with high marginal value to the solution will generate a significant amount of fitness *variance* upon perturbation. This variation makes it more difficult to detect and decide upon the correct values for less salient alleles, and often the continuation of the solution must wait until the type I error is corrected. Such an effect places a premium on perturbing or preserving diversity among only those alleles that have not yet been decided correctly.

With these ideas under our belts, we briefly review some of the theory necessary to perform first analyses of continuation economics and quality.

3 BUILDING BLOCK THEORY NEEDED HEREIN

Our theory needs are straightforward and have been available for some time. First, we need to estimate the relationship between solution quality and population size at the completion of a single epoch of a competent genetic algorithm. Second, we need to estimate the length of time to achieve such solutions. We start by examining the notion of a competent genetic algorithm, continue by considering rational population sizing, and continue by considering recent estimates of run duration.

3.1 COMPETENT GAs

Elsewhere (Goldberg, 1993), I have defined competent genetic algorithms as those that solve hard problems, quickly, reliably, and accurately. The ideal of competence has been approached in practice by a number of procedures, including the fast messy genetic algorithm (Goldberg, Deb, Kargupta, & Harik, 1993), the gene expression messy genetic algorithm (Kargupta, 1996), the linkage learning genetic algorithm (Harik, 1997; Harik & Goldberg, 1997), and the Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1999). In experimental studies, competent GAs can be modeled effectively by using one- or two-point crossover operators on codings where tight linkage has been prespecified (Goldberg, Deb, & Clark, 1992). In theoretical studies, the assumption of competence implies that the primary determinant of solution quality is adequate statistical decision making and thus population size, because the other conditions (building block growth, supply, mixing, and difficulty) are properly accounted.

3.2 RATIONAL POPULATION SIZING

Rational signal-to-noise population sizing was suggested in 1991 (Goldberg & Rudnick, 1991), tested in 1992 (Goldberg, Deb, & Clark, 1992), and refined in 1996 (Harik, Cantu-Paz, Goldberg, & Miller, 1996). The idea derives from Holland's idealization (Holland, 1973) of the decision making in genetic algorithms as multiple quasi-independent k-armed bandit problems. Although the k-armed bandit has undergone its share of criticism, the idea is profound and suggests that all decision-making in complex problems—even deterministic problems—is statistical in nature, because (1) hard problems force one to decompose, and (2) that decomposition means that one building block's experimentation is another building block's statistical variation or noise.

Here, we rely on recent work tying the solution of the GR problem to BB-wise decision making. Details of that work are available elsewhere (Harik, Cantu-Paz, Goldberg, & Miller, 1996). For our purposes we recognize that appropriate population sizing can be given approximately by an equation of the following form:

$$n = -c\frac{\sigma}{d}\sqrt{m}\ln\alpha, \tag{3}$$

where c depends on the complexity of the problem (constant for problems of given difficulty), α is the probability of not meeting criterion, σ is the root-mean-squared (RMS) building-block fitness variation, and d is the fitness accuracy or *signal* desired in the final solution. This equation will be used in two different ways in what follows.

3.3 RUN DURATION ESTIMATES

The other item we need in our calculations is run or epoch duration. An early study of takeover time can be used to estimate run durations of $O(\ell \log n)$ and convergence studies based on selection-intensity methods borrowed from quantitative genetics estimate durations of $O(\ell^{1/2})$ (Mühlenbein, 1992; Thierens & Goldberg, 1994) for problems with nearly uniformly scaled building blocks and of $O(\ell)$ for those with large variations in building block salience (Goldberg, 1997; Thierens, Goldberg, & Pereira, 1998). Since the duration is primarily determined by the size of the problem ℓ, for a given problem we consider epoch duration to be essentially constant as the population size varies.

With these three items as background, we now build and use two models of the economy of continuation in GEAs: a model of problems where building blocks are uniformly scaled and a model of problems where building blocks are badly (exponentially) scaled.

4 ECONOMY OF CONTINUATION, CASE I: UNIFORM SCALING

This section considers a problem where all building blocks are equally salient; that is, the proper solution of each building block results in an equal increment of fitness compared to that of any other. Additionally, we assume that the problem may not be solved easily by simple mutative GEAs or hillclimbers alone. Elsewhere (Goldberg, 1991), I have attempted to delineate between problems that are easy for simple hillclimbers and those that are not; we do not revisit those theories here. We do, however, recognize that any empirical test of these theories must be carried out with test functions that meet this assumption. More will be said about this later, but here we turn to the assemblage of our model.

4.1 ECONOMY BOTH IDEAL AND REAL

Under the assumption of discrete mixing epochs and an ICO, we may write the relationship between the number of function evaluations T, population size n, number of generations g, and epoch count e as follows:

$$T = egn \tag{4}$$

This equation assumes that there is no rework or other cost in transition from epoch to epoch. Moving from an idealized continuation operator to a real one, we consider the cost of rework as follows:

$$T = e[gn + r], \tag{5}$$

where r is the average rework in terms of numbers of function evaluations per epoch.

4.2 QUALITY AND POPULATION SIZE

These equations account properly for the economy of continuation in the idealized and real cases, but the solution quality must be evaluated by considering the population size, and we must make some assumption regarding the epoch duration. As discussed earlier, we will assume a constant number of generations per epoch (what we shall call g_0) irrespective of the variation in population size. Turning to the probability of not meeting criterion—what we shall call the probabilistic error or simply error—we appeal to the population sizing (equation 3) and write

$$-\ln\alpha_1 = \frac{n}{n_0}, \tag{6}$$

where α_1 is the error in the first epoch, and n_0 is a constant that depends on problem size, difficulty, signal-to-noise ratio, and building block chunk size (all of which are constant given a particular, fixed problem). Raising e to the indicated power on both sides of the equation yields

$$\alpha_1 = \exp(-n/n_0) \tag{7}$$

Under reasonable assumptions this leads to an interesting solution in the case of an idealized continuation operator.

4.3 SOLUTION FOR AN ICO

Imagine that the population from epoch to epoch is held constant and that an ICO is used to perturb only those variables that have been incorrectly solved. Thus, in the second epoch, we should expect that the error on the remaining improperly solved variables (call it the marginal error α'_2) should be less than or equal to that in the initial epoch: $\alpha'_2 \leq \alpha_1$. More generally, the marginal error of the ith generation should be less than the error of the initial generation.

$$\alpha'_i \leq \alpha_1, \quad i > 1 \tag{8}$$

Under conditions of ideal continuation then, we should expect the actual error in the eth epoch to go as the product of the actual marginal errors:

$$\alpha_e = \prod_{i=1}^{e} \alpha'_i \tag{9}$$

The exact solution of equation 9 is quite difficult and unnecessary for our purposes. The inequality of equation 8 allows us to bound the accumulated error of epoch e as

$$\alpha_e \leq \alpha_1^e \tag{10}$$

In the case of an idealized continuation operator, we may rewrite equation 4 as $e = T/(g_0 n)$ and we may then rewrite the inequality 10 as follows:

$$\alpha_e \leq [\exp(-n/n_0)]^{T/(g_0 n)} \tag{11}$$

The expression $[\exp(-n/n_0)]^{T/(g_0 n)}$ may be reduced to $\exp(-T/(g_0 n_0))$. Thus, we have shown that under idealized continuation the probability of solution error is less than a constant function of population size.

If we assume for the moment that marginal epoch error is equal to initial error (the uniform epoch error assumption), we note that genetic and evolutionary algorithms employing an ICO are *indifferent* to the epoch-size tradeoff: a single implicitly parallel solution costs the same as a solution solved building block by building block.

Next, we consider what happens if we hold the uniform epoch error assumption in place, but introduce the rework of a real continuation operator.

4.4 SOLUTION FOR AN RCO

If we permit the more realistic conditions of an RCO, we may express the number of epochs using equation 5 as $e = T/(ng_0 + r)$. and the accumulated error may be calculated as previously as

$$\alpha_e = \exp\left[\frac{-T/n_0}{g_0 + r/n}\right], \tag{12}$$

where the equality replaces the inequality under the uniform epoch error assumption. Inspection of the equation is enlightening. Assuming a fixed amount of time T and the ability to vary the population size n with all other values (including the rework) held as specified constants, minimization of the error suggests that the right thing to do is to choose as large a population size as possible. Indeed a small value of n causes the term r/n to be large, which makes the overall argument to the function exp a relatively small negative number, which makes the error relatively large. In practice we don't want to increase the population size so much as to not complete the run, and this condition leads to a solution in a single epoch ($e = 1$) with $n = T/g_0$.

Thus, under uniform scaling of the building blocks—under conditions of uniform salience—and the assumption of uniform epochal error, the economic solution is to solve the problem in a single epoch. This analysis places an economic face on the terms implicit parallelism for the first time. Although it may be argued, that the assumption of uniform epochal error is overly conservative, in real GEAs, the perturbation of RCOs will inevitably lead to the perturbation of building blocks with high marginal fitness, which in turn will lead to a need for large population sizes to promote effective continuation. More study of this tradeoff is warranted, but the analysis here is consonant with current knowledge about real GEAs. Moreover, the real need is to inquire about the economy of large versus small populations when the problem has a severely non-uniform salience distribution, a matter to be taken up in the next section.

5 ECONOMY OF CONTINUATION, CASE II: EXPONENTIAL SCALING

The straightforward calculation of the last section appears to suggest that continuation is a loser in problems of uniform or near-uniform salience. Here we modify the analysis to permit straightforward evaluation of the economy-quality tradeoff of continuation when building blocks are non-uniform in their contribution to the solution.

The economy equations (4 and 5) serve us well as before, but we need to take a somewhat different approach to assessing solution quality.

As before, we recognize that $\ln \alpha_1 = -n/n_0$, but here instead we define the coefficient n_0 for a problem with unit signal $d_0 = 1$, unit building-block RMS fitness variance $\sigma_0 = 1$, and a single building block $m_0 = 1$. Thus, in a problem where λ BBs are tackled in a given epoch with varying signals, the population sizing relationship may be written as

$$n = -n_0 \frac{\sqrt{\lambda}}{d} \ln \alpha_1. \tag{13}$$

As distinct from the uniform analysis, we set a constant reliability (and error) with $c_0 = -\ln \alpha_1$. Letting $n' = c_0 n_0$ yields

$$n = n' \frac{\sqrt{\lambda}}{d} \tag{14}$$

Thus, the population size required goes up as the square root of the number of BBs solved and inversely with the signal of the least salient BB among the candidate set.

5.1 SCALING RELATIONSHIPS

We now turn to the importance of the relative scaling of different building blocks. We assume that the signal d is a non-increasing function of the parameter λ. In words, λ is an index of variables or building blocks from high salience to low salience. This parameterization proved useful in a selection intensity solution of an exponentially scaled problem first presented elsewhere (Goldberg, 1997) and recently published in Thierens, Goldberg, & Pereira (1998).

5.2 EXPONENTIAL SCALING

We will generalize the result in a moment, but we start with an assumed exponentially decreasing BB signal

$$d = 2^{1-\lambda} \tag{15}$$

The bounding importance of the exponential case was recognized in Rudnick's (1992) and Thieren's (1995) theses, because when ordinal selection schemes are used it is the dividing line between problems where low salience building blocks can or cannot overpower a higher salience building block.

Substituting the signal relationship into the population equation 14, we obtain a relationship between population and salience rank as follows:

$$n = n' \sqrt{\lambda} 2^{\lambda - 1} \tag{16}$$

Substituting into the ICO economy equation (4) and rearranging in terms of the number of epochs.

$$e = \frac{T}{n' g_0 2^{\lambda-1} \sqrt{\lambda}} \tag{17}$$

If we now think of applying the equation iteratively under an ICO, the population size required in the next generation to correctly solve another λ BBs is the same as in the previous epoch (with appropriate rescaling). Thus, at the end of the run, the number of correctly solved building blocks will be the number per epoch (λ) times the number of epochs. Calling this quantity the *quality* Q yields

$$
\begin{aligned}
Q &= e\lambda && (18) \\
&= \frac{2T}{n' g_0} \frac{\lambda}{2^{\lambda} \sqrt{\lambda}} && (19) \\
&= c' f(\lambda) && (20)
\end{aligned}
$$

Ignoring the constant and recognizing that the function of λ decreases with increasing λ, the economic way to maximize quality in a badly scaled problem is to solve building block by building block.

This is remarkable, and is exactly the opposite of our conclusion under the uniformly salient building blocks. There, our reasoning suggested that a single implicitly parallel epoch was the way to go, whereas here, bad building block scaling makes it uneconomic to solve more than a single building block at a time. Clearly, there must be a dividing line between these two qualitatively different types of behavior. A matter that becomes clearer if we inspect the function of λ in the quality equation.

5.3 NEUTRAL SCALING

Noting that in the previous solution $f(\lambda) = \frac{\sqrt{\lambda}}{2^{\lambda}}$ and recognizing that the term 2^{λ} comes inversely from $d(\lambda)$, we might enquire as to what kind of function would make the quality function f indifferent to variations in λ. The dividing line occurs when $d = c\lambda^{-1/2}$: a function with salience that decreases in proportion to the inverse of the square root of the salience index.

Because of the fairly large number of assumptions made herein, we should not expect this prediction to be particularly crisp. Nonetheless, it should act as a rough guide to economics and quality interrelationships in real GEAs, and the next section suggests such study.

6 WHAT EVIDENCE AND WHAT'S NEXT?

This paper is largely a theoretical contribution, but we briefly cite a number of pieces of evidence to suggest that the theory is at least qualitatively correct and outline a research program to test the theories more rigorously.

6.1 4 PIECES OF EVIDENCE

There are at least four pieces of evidence to support the theories herein:

1. Results on uniformly scaled order-k trap functions with a 1+1 ES (Mühlenbein, 1992) come more slowly than with a competent GA on the same problems (Goldberg, Deb, Kargupta, & Harik, 1993).
2. Empirically, micro-GAs are often successful in real problems (KrishnaKumar, 1989).
3. Adaptive and self-adaptive ESs are preferred to fixed ESs, and adaptive and self-adaptive ESs with recombination are preferred to their counterparts without (Bäck & Schwefel, 1995).
4. The linkage learning GA can more easily solve badly scaled problems than uniformly scaled ones (Harik, 1997)

These items deserve additional discussion, and an expanded paper will do just that at a later date. At this time, interested readers should consult the relevant literature, and consider the proposed experimental program outlined below.

6.2 AN EXPERIMENTAL PROGRAM

The models of this paper make a number of predictions that can be confirmed through careful experiments. Experiments are being designed with the following constraints in mind.

- The theory assumes the operation of competent mixing and continuation.
- Problems that may be solved bitwise or with continuation (selection and mutation) alone will not demonstrate the utility of mixing.
- Poorly designed mixing operators will not demonstrate the utility of mixing.
- Poorly designed continuation operators will not demonstrate the utility of continuation.

Competent mixing operators can be approximated with tight linkage and low-order or tailored crossing operators, and an ICO can be modeled by using problem knowledge and always mutating improperly converged alleles at relatively high rates.

7 CONCLUSIONS

This paper has considered the economy of continuation and has attempted to draw a sharper line between those problems in which substantial implicit parallelism should be undertaken and those where a more serial mode of processing should be adopted. The dividing line and launch point for the discussion largely comes from thinking of the decision making in GEAs as largely being statistical in nature and this helps us to make some rational distinctions between implicitly parallel versus serial processing.

An economic role for effective mixing has been outlined and likewise a role for effective continuation has been suggested. Some limited empirical evidence has been cited for the theory qualitatively, and more systematic experiment is clearly needed and has been outlined. But if the predictions of the paper hold up, the great debate between crossover and mutation may soon be replaced with a new respect for each of their important—and economic—roles in obtaining high quality solutions to the array of problems before us.

Acknowledgments

This paper was conceived while I was on sabbatical at Dortmund University. I am grateful to H.-P. Schwefel for inviting me to Dortmund. I also thank Jacob Borgerson for drawing the figures.

My work on this paper was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-97-1-0050. Research funding was also provided by the U. S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0003. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

The views and conclusions contained herein are those of the author and should not be interpreted as necessarily representing the official policies or endorsements of the Air Force Office of Scientific Research, the U. S.

Army Research Laboratory, or the U. S. Government.

References

Bäck, T. & Schwefel, H.-P. (1995). Evolution Strategies I: Variants and their computational implementation. In G. Winter, J. Périaux, M. Galán, & P. Cuesta, *Genetic algorithms in engineering and computer science* (pp. 111–126). Chichester: John Wiley.

Goldberg, D. E. (1991). Real-coded genetic algorithms, virtual alphabets, and blocking. *Complex Systems, 5*, 139–167. (originally published as IlliGAL 90001)

Goldberg, D. E. (1993). Making genetic algorithms fly: A lesson from the Wright Brothers. *Advanced Technology for Developers, 2* 1–8.

Goldberg, D. E. (1997). *Estimating central building block time.* Unpublished manuscript.

Goldberg, D. E., Deb, K., & Clark, J. (1992). Genetic algorithms, noise, and the sizing of populations. *Complex Systems, 6*, 333–362.

Goldberg, D. E., Deb, K., Kargupta, H., & Harik, G. (1993). Rapid accurate function optimization using fast messy genetic algorithms. *Proceedings of the Fifth International Conference on Genetic Algorithms,* 56–64.

Goldberg, D. E., & Rudnick, M. (1991). Genetic algorithms and the variance of fitness. *Complex Systems, 5*, 265–278.

Goldberg, D. E., & Smith, R. E. (1987). Nonstationary function optimization using genetic algorithms with dominance and diploidy. *Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms,* 59–68.

Harik, G. R. (1997). *Learning gene linkage to efficiently solve problems of bounded difficulty using genetic algorithms* (PhD Dissertation, University of Michigan & IlliGAL Report No. 97005). Urbana, IL: Illinois Genetic Algorithms Laboratory, University of Illinois at Urbana-Champaign.

Harik, G. R., Cantú-Paz, E., Goldberg, D. E. and Miller, B. L. (1997). The gambler's ruin problem, genetic algorithms, and the sizing of populations, *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation,* 7–12.

Harik, G. R., & Goldberg, D. E. (1997). Learning linkage. *Foundations of Genetic Algorithms, IV*, 247–262.

Holland, J. H., (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing, 2*(2), 88–105.

Kargupta, H. (1996). *SEARCH, evolution, and the gene expression messy genetic algorithm* (Report No. LA-UR-96-60). Los Alamos: Los Alamos National Laboratory.

KrishnamKumar, K. (1989). Microgenetic algorithms for stationary and nonstationary function optimization. *SPIE Proceedings on Intelligent Control and Adaptive Systems,* 289–296.

Mühlenbein, H. (1992). How genetic algorithms really work I: Mutation and hillclimbing. *Parallel Problem Solving from Nature, 2*, 15–26.

van Nimwegen, E., & Crutchfield, J. P. (in press). Optimizing epochal evolutionary search: Population-size independent theory. *Computer Methods in Applied Mechanics and Engineering*

Pelikan, M., Goldberg, D. E., & Cantú-Paz (1999). BOA: The Bayesian optimization algorithm. *GECCO-99: Proceedings of the 1999 Genetic and Evolutionary Computation Conference* (elsewhere in this volume).

Rudnick, M. (1992). *Genetic algorithms and fitness variance with an application to the automated design of artificial neural networks* (PhD Dissertation). Beaverton, OR: Computer Science and Engineering Department, Oregon Graduate Institute.

Thierens, D. (1995). *Analysis and design of genetic algorithms* (PhD dissertation). Leuven, Belgium: Electrical Engineering Department, Catholic University of Leuven.

Thierens, D., & Goldberg, D. E. (1994). Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving from Nature, 2*, 119–129.

Thierens, D., Goldberg, D. E., & Pereira, A. (1998). Domino convergence, drift, and the temporal-salience structure of problems. *1998 IEEE International Conference on Evolutionary Computation Proceedings,* 535–540.