Let's Get Ready to Rumble Redux: Crossover Versus Mutation Head to Head on Exponentially Scaled Problems

Kumara Sastry
Illinois Genetic Algorithms Laboratory (IlliGAL)
Industrial and Enterprise Systems Engineering
Materials Computation Center
University of Illinois at Urbana-Champaign
Urbana IL 61801
ksastry@uiuc.edu

David E. Goldberg
Illinois Genetic Algorithms Laboratory (IlliGAL)
Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana IL 61801
deg@uiuc.edu

ABSTRACT

This paper analyzes the relative advantages between crossover and mutation on a class of deterministic and stochastic additively separable problems with substructures of non-uniform salience. This study assumes that the recombination and mutation operators have the knowledge of the building blocks (BBs) and effectively exchange or search among competing BBs. Facetwise models of convergence time and population sizing have been used to determine the scalability of each algorithm. The analysis shows that for deterministic exponentially-scaled additively separable, problems, the BB-wise mutation is more efficient than crossover yielding a speedup of $o(\ell \log \ell)$, where ℓ is the problem size. For the noisy exponentially-scaled problems, the outcome depends on whether scaling on noise is dominant. When scaling dominates, mutation is more efficient than crossover yielding a speedup of $o(\ell \log \ell)$. On the other hand, when noise dominates, crossover is more efficient than mutation yielding a speedup of $o(\ell)$.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Genetic algorithms, scaling, exponential scaling, salience, crossover, mutation, population sizing, convergence time, scalability analysis, domino convergence, drift time, building blocks, noisy fitness functions, efficiency enhancement, speedup

GECCO'07, July 7–11, 2007, London, England, United Kingdom.

1. INTRODUCTION

Great debate between crossover and mutation has consumed much ink and many trees over the years. When mutation works it is lightening quick and uses small or non-extent populations. Crossover when it works, seems to be able to tackle more complex problems, but getting the population size and other parameters set is a challenge. Recently, Sastry and Goldberg [33] presented an unbiased comparison between the scalability of crossover and mutation on a class of uniformly-scaled additively-separable problems with and without additive Gaussian noise. In this paper we extend the analysis to an important class of non-uniformly scaled additively-separable problems with and without additive Gaussian noise.

Assuming that both the recombination and mutation operators possess linkage (or neighborhood) knowledge, we pit them against each other for solving exponentially-scaled boundedly-difficult additively-separable problems with and without the presence of additive exogenous noise. We use a recombination operator that exchanges building blocks (BBs) without disrupting them and a mutation operator that performs local search among competing building-block neighborhood. The motivation for this study also comes from recent local-search literature, where authors have highlighted the importance of using a good *neighborhood* operator [6, 16, 29, 41]. However, a systematic method of designing a good neighborhood operator for a class of search problems is still an open question. We investigate whether using a neighborhood operator that searches among competing BBs of a problem would be advantageous and if so under what circumstances.

This paper is organized as follows. The next section gives a brief review of related literature. We provide an outline of the crossover-based and mutation-based genetic algorithms (GAs) in Section 3. Facetwise models are developed to determine the scalability of the crossover and the BB-wise mutation-based GAs for deterministic fitness functions in Section 4 and for noisy fitness functions in Section 5. Finally, we provide summary and conclusions.

2. LITERATURE REVIEW

Over the last few decades many researchers have empirically and theoretically studied where genetic algorithms excel. An exhaustive review is beyond the scope of this paper, and therefore we present a brief review of related studies.

Several authors have analyzed the scalability of a mutation based hillclimber and compared it to scalability of different forms of genetic algorithms [2, 7, 9, 19, 20, 26, 27, 28, 38]. Goldberg [11] gave a facetwise analysis of deciding between a single run with a large population GA and multiple runs with several small population GAs, under the constraint of fixed computational cost. He showed that for uniformly-scaled problems a single run of large population GA was advantageous, while for exponentially-scaled problems small population GAs with multiple restarts were better. Srivastava and Goldberg [37, 36] empirically verified and analytically enhanced the *time-continuation* theory put forth by Goldberg [11]. Recently, Cantú-Paz and Goldberg [5] investigated scenarios under which multiple runs of a GA are better than a single GA run. For an exhaustive review of studies on the advantages/disadvantages of multiple populations both under serial and parallel GAs over a single large-population GA, the reader is referred elsewhere [4, 36, 23, 8] and to the references therein.

While many of the related studies [11, 37, 5] assumed fixed genetic operators, with no knowledge of building-block structure, the authors [33] assumed that the recombination and mutation operators have linkage (or neighborhood) knowledge. We showed that for uniformly-scaled, additively separable search problems with deterministic fitness functions, building-block-wise mutation provided a speed-up of $o(k \log m)$)—where k is the building block size, and m is the number of building blocks—over recombination. On the other hand, for uniformly-scaled, additively separable search problems with additive Gaussian noise, building-block-wise recombination provided a speedup of $o(m\sqrt{k}/\log m)$. Based on this study methods for inducing neighborhoods for a scalable mutation operator have been proposed that demonstrated polynomial (usually subquadratic) scalability on uniformly-scaled additively separable problems [32, 22]. Recently, Sastry et al. [35] considered fluctuating crosstalk or non-linear interactions of building blocks [12] and showed that cross talk behaved like exogenous noise and recombination provided speed-up over mutation until the strength of the crosstalk far exceeds the underlying fitness variance.

In this study, we follow our earlier approach [33] of assuming that both recombination and mutation operators have knowledge of building blocks of the underlying search problem and extending the analysis to a class of non-uniformly-scaled additively decomposable problems with and without additive Gaussian noise.

3. PRELIMINARIES

The objective of this paper is to predict the relative computational costs of a crossover and an ideal-mutation based algorithm for exponentially-scaled additively separable problems with and without additive Gaussian noise. Before developing models for estimating the computational costs, we briefly describe the algorithms and the assumptions used in the paper.

3.1 Selectorecombinative Genetic Algorithms

We consider a generationwise selectorecombinative GA with non-overlapping populations of fixed size [18, 10]. We apply crossover with a probability of 1.0 and do not use any mutation. We assume binary strings of fixed length as the chromosomes. To ease the analytical burden, the selection mechanism assumed throughout the analysis is binary tournament selection [14]. However, the results can be extended to other tournament sizes and other selection methods in a straightforward manner. The recombination method used in the analysis is a uniform building-block-wise crossover [39]. In uniform BB-wise crossover, two parents are randomly selected from the mating pool and their building blocks in each partition are exchanged with a probability of 0.5. Therefore, none of the building blocks are disrupted during a recombination event. The offspring created through crossover entirely replace the parental individuals.

3.2 Building-Block-Wise Mutation Algorithm (BBMA)

In this paper we consider an *enumerative BB-wise mutation* operator, in which we start with a random individual and evaluate all possible schemas in a given partition. That is, for a building-block of size k, we evaluate all 2^k individuals. The best out of 2^k individuals is chosen as a candidate for mutating BBs of other partitions. In other words, the BBs in different partitions are mutated in a sequential manner. For a problem with m BBs of size k each, the BBMA can be described as follows:

1. Start with a random individual and evaluate it.

2. Consider the first non-mutated BB. Here the BB order is chosen arbitrarily from left-to-right, however, different schemes can be—or may required to be—chosen to decide the order of BBs.

3. Create $2^k - 1$ unique individuals with all possible schemata in the chosen BB partition. Note that the schemata in other partitions are the same as the original individual (from step 2).

4. Evaluate all $2^k - 1$ individuals and retain the best for mutation of BBs in other partitions.

5. Repeat steps 2–4 till BBs of all the partitions have been mutated.

We use an enumerative BB-wise mutation for simplifying the analysis and a greedy BB-wise method can improve the performance of the mutation-based algorithm. A straightforward Markov process analysis—along the lines of [27, 28]—of a greedy BB-wise mutation algorithm indeed shows that the greedy method is on an average better than the enumerative one. However, the analysis also shows that differences between the greedy and enumerative BB-wise mutation approaches are little, especially for moderate-to-large problems. Moreover, the computational costs of an enumerative BB-wise mutation bounds the costs of a greedy BB-wise mutation.

3.3 Test Problem

Our approach in testing cGA and other search methods is to consider problems from a *design envelope* perspective and to follow a Cartesian decomposition of different facets of problem difficulty [12]. Here we consider two facets of problem difficulty: scaling and noise. As a representative of badly-scaled noisy problem, we consider the noisy BinInt problem [30, 40], where the objective is to maximize an unsigned binary-integer function with or without the presence of additive Gaussian noise of specified variance, σ_N^2,

$$f(\mathbf{x}) = \sum_{j=1}^{\ell} 2^{j-1} x_j + \mathcal{N}\left(0, \sigma_N^2\right), \tag{1}$$

where ℓ is the problem size.

4. CROSSOVER VS. MUTATION: DETERMINISTIC FITNESS FUNCTIONS

In this section we analyze the relative computational costs of using a selectorecombinative GA or a BB-wise mutation algorithm for successfully solving exponentially-scaled deterministic problems of bounded difficulty. The objective of the analysis is to answer whether a population-based selectorecombinative GA is computationally advantageous over a BB-wise-mutation based algorithm. If one algorithm is better than the other, we are also interested in estimating the savings in computational time. Note that unlike earlier studies, we assume that the building-block structure is known to both the crossover and mutation operators.

We begin our analysis with the scalability of selectorecombinative genetic algorithms followed by the scalability of the BB-wise mutation algorithm.

4.1 Scalability of Selectorecombinative GA

Two key factors for predicting the scalability and estimating the computational costs of a genetic algorithm are the convergence time and population sizing. Therefore, in the following subsections we present facetwise models of convergence time and population sizing.

4.1.1 Convergence-Time Model

When dealing with non-uniformly scaled problems, GAs pay attention to the most salient building block first, a condition sometimes called *domino convergence* [30]. Thierens, Goldberg, & Pereira [40] used the domino-convergence parameterization and proposed a convergence time model for selectorecombinative GAs for the BinInt problem:

$$t_c = c_c \cdot \ell, \tag{2}$$

where, $c_c = \sqrt{3}\log 2/I$, and I is the selection intensity [3]. For binary tournament selection, $I = 1/\sqrt{\pi}$.

4.1.2 Population-Sizing Model

Goldberg, Deb, & Clark [13] proposed population-sizing models for correctly deciding between competing BBs. They incorporated noise arising from other partitions into their model. However, they assumed that if wrong BBs were chosen in the first generation, the GAs would be unable to recover from the error. Harik, Cantú-Paz, Goldberg, and Miller [17] refined the above model by incorporating cumulative effects of decision making over time rather than in first generation only. Harik et al. [17] modeled the decision making between competing BBs as a gambler's ruin problem which showed that the population sizing for ensuring correct decision making and building-block supply scales as $o(\frac{\sigma_{BB}}{d} 2^k \sqrt{m}\log m)$.

However, for exponentially-scaled problems *genetic drift* plays a critical, and often dominating, role in the performance of selectorecombinative GAs and the population size has to be sized to circumvent drift [40, 12]. From the genetic drift models [21, 15, 1], we know that the relation between

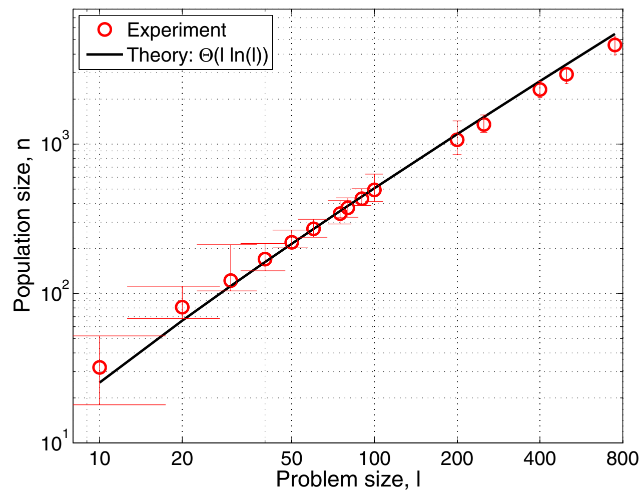

Figure 1: Empirical verification of the population-sizing required by selectorecombinative GA for the BinInt problem. The results follow the theoretical predictions of Equation 4. The empirical results are averaged over 30 independent bisection runs. The results show that the population size scales as $o(\ell \log \ell)$.

drift time—defined as the number of generations required to converge to a solution purely due to drift—as,

$$t_d = c_d \cdot n \tag{3}$$

where t_d is the drift time, c_d is a constant which is usually equal to 1.4, and n is the population size. Since we want to avoid the genetic drift and would would like to have a probabilistic safety factor of correctly converging on at least $\ell - 1$ out of ℓ BBs. Therefore, we should size the population such that $t_d > t_c \log \ell$:

$$n = \frac{c_c}{c_d} \ell \log \ell. \tag{4}$$

Therefore, for exponentially-scaled problems the population size scales as $o(\ell \log \ell)$. The above population-sizing model is empirically verified in Figure 1. The minimum population size was determined by a bisection method [31] where the solution quality for each bisection iteration was averaged over 50 independent GA runs and the population size reported is average of 30 such bisection runs. The results show that the experiments follow theoretical prediction.

Using equations 4 and 2, we can now predict the scalability, or the number of function evaluations required for successful convergence, of GAs as follows:

$$n_{\text{fe,GA}} = n \cdot t_c = c_{fe} \ell^2 \log \ell, \tag{5}$$

where $c_{fe} = c_c^2/c_d$. The above theoretical model for the scalability of the selectorecombinative GA is empirically verified in Figure 2. The results are averaged over 900 independent runs and follow theoretical prediction.

4.2 Scalability of BB-wise Mutation Algorithm

Since the initial point is evaluated once and after that for each of the m BBs, $2^k - 1$ individuals are evaluated, the total

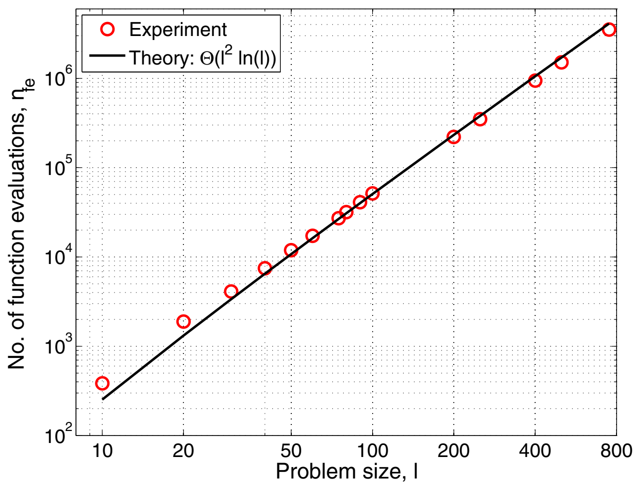

Figure 2: Empirical verification of the scalability of the selectorecombinative GA for the BinInt problem. The results follow the theoretical predictions of Equation 5. The empirical results are averaged over 1500 independent runs. The results show that the number of function evaluations scales as $o(\ell^2 \log \ell)$.

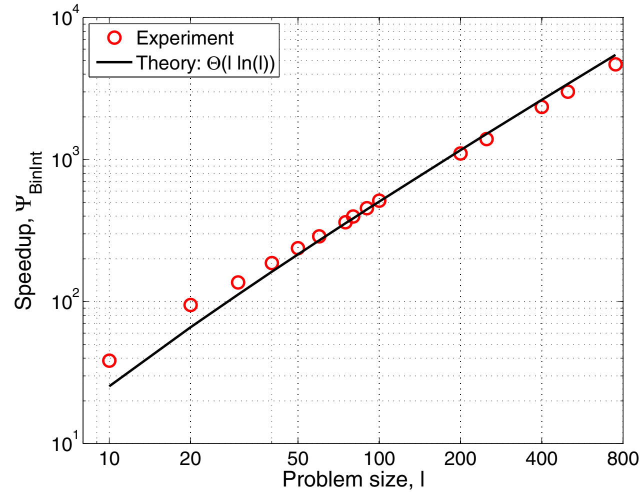

Figure 3: Empirical verification of the speed-up predicted for using BB-wise mutation over a selectorecombinative GA by Equation 7 on the deterministic exponentially-scaled problem. The empirical results are averaged over 1500 independent runs. The results show that the speed-up obtained by BB-wise mutation algorithm over a GA is $o(\ell \log \ell)$.

number of function evaluations required for the BBMA is

$$n_{\text{fe,BBMA}} = \left(2^k - 1\right) m + 1. \tag{6}$$

For the BinInt problem, $k = 1$ and $m = \ell$.

The results from the above subsections (Equations 5 and 6) indicate that while the scalability of a selectorecombinative GA is $o\left(\ell^2 \log \ell\right)$, the scalability of the BBMA is $o\left(\ell\right)$. By searching among building-block neighborhoods, the selectomutative algorithm scales-up significantly better than a mutation operator with no linkage information and provides a savings of $o(\ell \log \ell)$ evaluations over a selectorecombinative GA. This savings is expected and has been observed by earlier studies [11, 37, 36] comes because the exponential scaling induces sequential processing of the building blocks as opposed to parallel processing in uniformly-scaled problems.

The speed-up—which is defined as the ratio of number of function evaluations required by a GA to that required by BBMA—obtained by using a BB-wise mutation algorithm over a selectorecombinative GA is given by

$$\eta = \frac{n_{\text{fe,GA}}}{n_{\text{fe,BBMA}}} = c_{fe} \ell \log \ell = o\left(\ell \log \ell\right). \tag{7}$$

The speed-up predicted by Equation 7 is verified with empirical results in Figure 3. The results are averaged over 1500 independent GA runs. The results show that there is a good agreement between the predicted and observed speed-up. The results show that for deterministic additively separable problems with exponentially-scaled BBs, a BB-wise mutation algorithm is about $o(\ell \log \ell)$ faster than a selectorecombinative GA.

5. CROSSOVER VS. MUTATION: NOISY FITNESS FUNCTIONS

In the previous section, we observed that BB-wise mutation scales-up better than a crossover on deterministic additively separable problems with exponentially-scaled building blocks. In this section we introduce another dimension of problem difficulty in *extra-BB noise* [12] and analyze if the BB-wise mutation maintains its edge over crossover. That is, we analyze whether a selectorecombinative or a selectomutative GA works better on additively separable problems with exponentially-scaled building blocks and with additive external Gaussian noise.

We follow the same approach outlined in the previous section and consider the scalability of crossover and mutation.

5.1 Scalability of Selectorecombinative GAs

Again we use the convergence-time and population-sizing models to determine the scalability of GAs under the presence of unbiased Gaussian noise for exponentially-scaled problems. Here we set the exogenous noise variance in relation to the initial deterministic fitness variance. That is $\sigma_N^2 = \rho_x \sigma_{f,\max}^2$, where $\sigma_{f,\max}^2 \approx 2^{\ell-1}/\sqrt{12}$ is the deterministic fitness variance of the initial population [40]. Therefore $\sigma_N^2 \approx \rho_x 2^{\ell-1}/\sqrt{12}$. In the presence of exogenous noise, there are two regimes: (1) scaling-dominated regime, $\sigma_N^2 \ll \sigma_f^2$, and (2) noise-dominated regime, $\sigma_N^2 \gg \sigma_f^2$, and we present models for both in the following paragraphs.

5.1.1 Convergence-Time Model

The convergence time for the scaling-dominated regime is given by Equation 2. For the noise-dominated regime, we use an approximate form of convergence-time model proposed by Miller and Goldberg [25]:

$$t_c = c_c' \sigma_f^2 + \sigma_N^2 \approx c_c' \rho_x 2^{\ell}, \tag{8}$$

where $c_c' = \pi/2I$. A detailed derivation of the above equation and other approximations are given elsewhere [12, 31].

Therefore, the convergence time of selectorecombinative

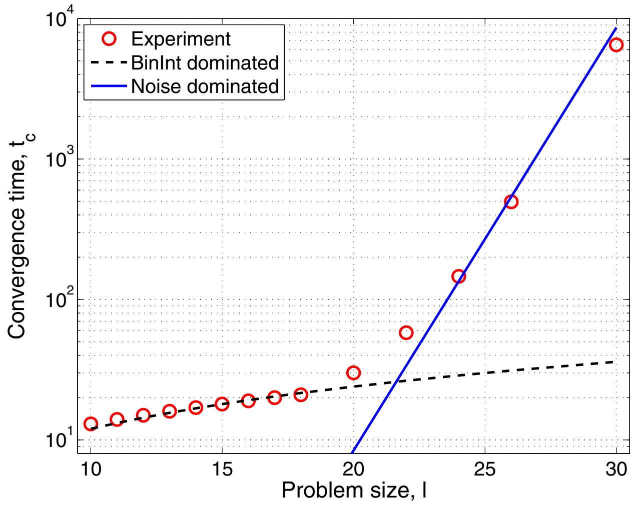

Figure 4: Empirical verification of the convergence-time models for the selectorecombinative GA for the noisy BinInt problem with $\rho_x = 10^{-5}$. The results follow the theoretical predictions of Equation 9. The empirical results are averaged over 1500 independent runs.

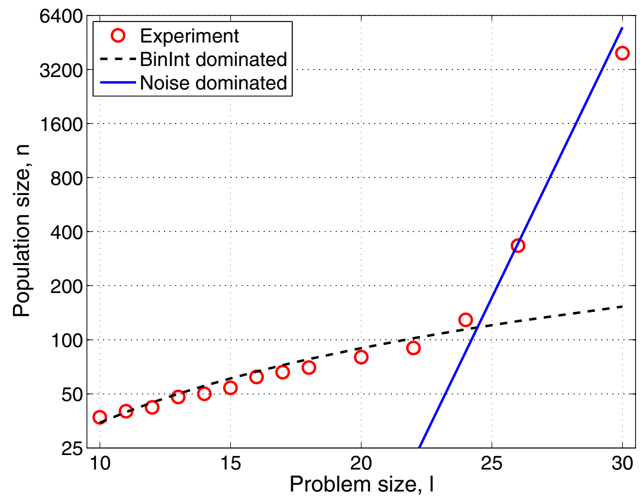

Figure 5: Empirical verification of the population-sizing required by selectorecombinative GA for the noisy BinInt problem with $\rho_x = 10^{-5}$. The results follow the theoretical predictions of Equation 11. The empirical results are averaged over 30 independent bisection runs.

GA for a noisy BinInt problem is given by

$$t_c = \begin{cases} c_c \ell & \sigma_N^2 \ll \sigma_f^2 \\ c_c' \rho_x 2^\ell & \sigma_N^2 \gg \sigma_f^2 \end{cases} . \quad (9)$$

The empirical validation of the above model is shown in Figure 4 for $\rho_x = 10^{-5}$. We have tried other values of ρ_x and the results are qualitatively similar and are shown elsewhere [34]. The results follow theoretical predictions and show that regions where noise-dominated and scale-dominated model apply.

5.1.2 *Population-Sizing Model*

The population size for the scaling-dominated regime is given by Equation 4. For the noise-dominated regime, the population size is given by the gambler's ruin model [17]. An approximate form of the gambler's ruin population-sizing model for noisy environments is given by

$$n = c_n \frac{\sigma_{f,\max}^2 + \sigma_N^2}{d_{\min}}, \quad (10)$$

where $c_n = \sqrt{pi}$, $d_{\min}$ is the minimum signal difference between the competing BBs [24, 12]. For the BinInt problem, $d_{\min} = 1$.

Therefore, the population sizing for the selectorecombinative GA for a noisy BinInt problem is given by

$$n = \begin{cases} \frac{c_c}{c_d} \ell \log \ell & \sigma_N^2 \ll \sigma_f^2 \\ c_n \rho_x 2^\ell & \sigma_N^2 \gg \sigma_f^2 \end{cases} . \quad (11)$$

Figure 5 depicts the empirical validation of the above population-sizing model for $\rho_x = 10^{-5}$. The results for other values of ρ_x are given elsewhere [34]. The results follow theoretical predictions and show that regions where noise-dominated and scale-dominated model apply.

Using equations 4 and 2, we can now predict the scalability, or the number of function evaluations required for successful convergence, of GAs for both noise-dominated and scaling-dominated regimes, as follows:

$$n_{\text{fe,GA}} = \begin{cases} c_{fe} \ell^2 \log \ell & \sigma_N^2 \ll \sigma_f^2 \\ c_{fe}' \rho_x^2 2^{2\ell} & \sigma_N^2 \gg \sigma_f^2 \end{cases} . \quad (12)$$

The empirical validation of the above model is shown in Figure 6 for $\rho_x = 10^{-5}$. We have tried other values of ρ_x and the results are shown elsewhere [34]. The results follow theoretical predictions and show that regions where noise-dominated and scale-dominated model apply.

5.2 Scalability of BB-wise Mutation Algorithm

Unlike the deterministic case where a BB was perturbed and evaluated once, in the noise-dominated regime we cannot rely on only a single evaluation. In other words, in the presence of noise, an average of multiple samples of the fitness should be used in deciding between competing building blocks. The number of samples required for evaluating the average fitness is given by [33]:

$$n_s = 2c\sigma_N^2, \quad (13)$$

where n_s is the number of independent fitness samples, and c is the square of the ordinate of a one-sided standard Gaussian deviate at a specified error probability α. Here we use $\alpha = 1/m$.

Since the initial point is evaluated n_s times and after that for each of the m BBs, $2^k - 1$ individuals are evaluated n_s times, the total number of function evaluations required for the BBMA for noisy fitness functions is given by

$$\begin{aligned} n_{\text{fe,BBMA}} &= n_s \left[\left(2^k - 1 \right) m + 1 \right], \\ &= c_{fe,m} \rho_x^2 2^{2\ell} \ell. \end{aligned} \quad (14)$$

where $c_{fe,m} = 2c/\sqrt{12}$ is a constant.

The scalability of selectorecombinative GA is compared to that of selectomutative GA for the noisy BinInt problem

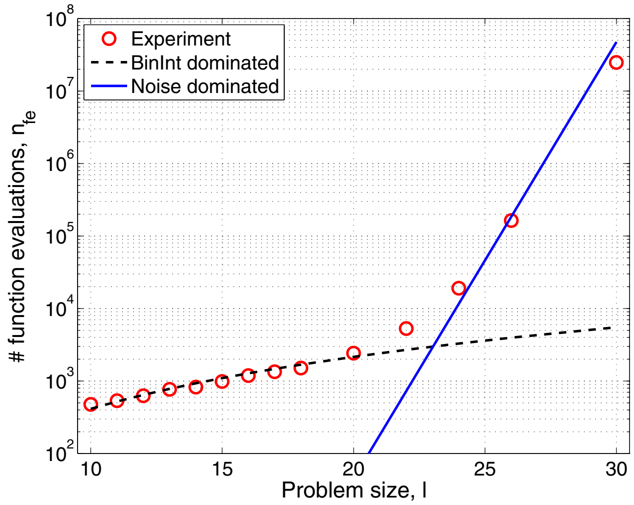

Figure 6: Empirical verification of the scalability of the selectorecombinative GA for the noisy BinInt problem with $\rho_x = 10^{-5}$. The results follow the theoretical predictions of Equation 12. The empirical results are averaged over 1500 independent runs.

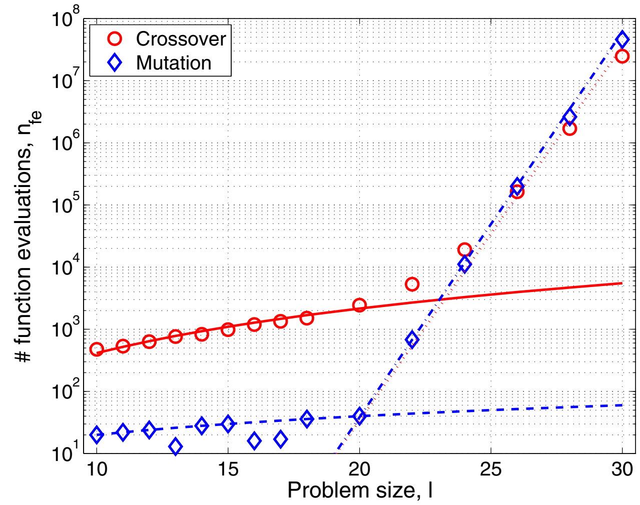

Figure 7: Comparison of scalability of selectorecombinative GA and selectomutative GA for the noisy BinInt problem with $\rho_x = 10^{-5}$. The results follow theoretical predictions of Equations 12 and 14. The empirical results are averaged over 1500 independent runs.

with $\rho_x = 10^{-5}$ in Figure 7. Comparisons for other values of ρ_x is shown elsewhere [34]. The figures show that the empirical results follow theoretical predictions. The results from the above subsections (Equations 12 and 14) indicate that under the presence of exogenous noise, a selectorecombinative GA scales as $o\left(\ell^2 \log \ell\right)$ in the scale-dominated regime and $o\left(\rho_x^2 2^{2\ell}\right)$ in the noise-dominated regime. On the other hand, the BB-wise mutation scales as $o(\ell)$ in the scale-dominated regime and $o\left(\rho_x^2 2^{2\ell}\ell\right)$ in the noise-dominated regime. Therefore, in the scale-dominated regime, the BB-wise mutation is $o(\ell \log \ell)$ and in the noise-dominated regime, a selectorecombinative GA is $o(\ell)$ times faster than the BB-wise mutation. By implicitly averaging out the exogenous noise, crossover is able to overcome the extra effort needed for the convergence and decision-making. On the other hand the explicit averaging via multiple fitness samples by the BB-wise mutation leads to an order of magnitude increase in the number of function evaluations.

The speed-up—which is defined as the ratio of number of function evaluations required by mutation to that required by crossover—obtained by using a selectorecombinative over selectomutative GA for the noisy BinInt problem is given by

$$\eta_{\text{Noise}} = \frac{n_{\text{fe,BBMA}}}{n_{\text{fe,GA}}} = \begin{cases} \frac{1}{c_{fe}\ell \log \ell} & \sigma_N^2 \ll \sigma_f^2 \\ \frac{c_{fe,m}\ell}{c'_{fe}} & \sigma_N^2 \gg \sigma_f^2 \end{cases}. \quad (15)$$

The speed-up predicted by Equation 15 is verified with empirical results in Figure 8 for $\rho_x = 10^{-5}$. The results for other values of ρ_x are shown elsewhere [34]. The results are averaged over 1500 independent runs. The results show that there is a good agreement between the predicted and observed speed-up. The results show that for stochastic, exponentially-scaled additively-separable problems, the efficiency of recombination and mutation depends on the dominating regime. In scale-dominated regime, BB-wise mutation algorithm is more efficient than crossover yielding a speedup of $o(\ell \log \ell)$. On the other hand, in noise-dominated regime, crossover is more efficient than mutation, yielding a speedup of $o(\ell)$.

6. SUMMARY & CONCLUSIONS

In this paper, we pitted crossover and mutation on a class of non-uniformly scaled, additively decomposable problems with and without additive Gaussian noise. In this study we assumed that both crossover and mutation have the knowledge of the important building blocks required to solve the problem. We considered ideal recombination and mutation operators, where the recombination operators exchanges building blocks without disruption and the mutation operators searches for the best building block in the building-block neighborhood.

We compared the computational costs BB-wise mutation algorithm with a selectorecombinative genetic algorithm for both deterministic and stochastic additively separable problems. Our results show that the BB-wise mutation provides significant advantage over crossover for deterministic problems with exponentially scaled problems yielding a speedup of $o(\ell \log \ell)$, where ℓ is the problem size. For noisy, exponentially-scaled problems, the outcome is mixed depending on whether noise is dominating or the scale. For scale-dominated problems, mutation is more efficient than crossover yielding a speedup of $o(\ell \log \ell)$. However, for the noise-dominated region, crossover is more efficient than mutation yielding a speedup of $o(\ell)$.

This study advances earlier studies that considered the relative advantages of crossover and mutation on uniformly-scaled problems and problems with non-linear interactions between building blocks and forms another building block in developing a theory of *time continuation*. The models and the results presented in this study can lead us to develop *adaptive time continuation operators* that can automatically identify the problem regime and choose the more efficient combination of operators.

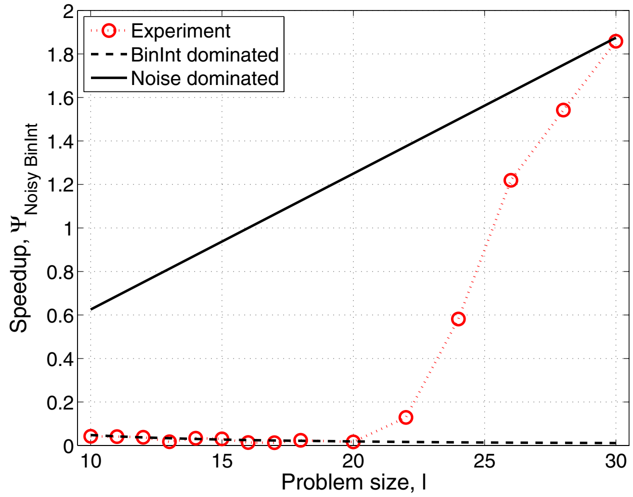

Figure 8: Empirical verification of the speed-up predicted for using BB-wise mutation over a selectorecombinative GA by Equation 15 for the BinInt problem with exogenous noise with $\rho_x = 10^{-5}$. The empirical results are averaged over 1500 independent runs.

Acknowledgments

This work was also sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant FA9550-06-1-0096, the National Science Foundation under ITR grant DMR-03-25939 at the Materials Computation Center. 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 authors 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.

7. REFERENCES

[1] H. Asoh and H. Mühlenbein. On the mean convergence time of evolutionary algorithms without selection and mutation. *Parallel Problem Solving from Nature*, 3:98–107, 1994.

[2] E. B. Baum, D. Boneh, and C. Garrett. Where genetic algorithms excel? *Evolutionary Computation*, 9(1):93–124, 2001.

[3] M. G. Bulmer. *The Mathematical Theory of Quantitative Genetics*. Oxford University Press, Oxford, 1985.

[4] E. Cantú-Paz. *Efficient and accurate parallel genetic algorithms*. Kluwer Academic Pub, Boston, MA, 2000.

[5] E. Cantú-Paz and D. E. Goldberg. Are multiple runs of genetic algorithms better than one? *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 801–812, 2003.

[6] B. W. Colletti and J. W. Barnes. Using group theory to construct and characterize metaheuristic search neighborhoods. In C. Rego and B. Alidaee, editors, *Adaptive Memory and Evolution: Tabu Search and Scatter Search*, pages 303–329. Kluwer Academic Publishers, Boston, MA, 2004.

[7] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276:51–81, 2002.

[8] M. Fuchs. Large populations are not always the best choice in genetic programming. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1033–1038, 1999.

[9] J. Garnier, L. Kallel, and M. Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):173–203, 1999.

[10] D. E. Goldberg. *Genetic algorithms in search optimization and machine learning*. Addison-Wesley, Reading, MA, 1989.

[11] D. E. Goldberg. Using time efficiently: Genetic-evolutionary algorithms and the continuation problem. *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pages 212–219, 1999. (Also IlliGAL Report No. 99002).

[12] D. E. Goldberg. *Design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.

[13] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992. (Also IlliGAL Report No. 91010).

[14] D. E. Goldberg, B. Korb, and K. Deb. Messy genetic algorithms: Motivation, analysis, and first results. *Complex Systems*, 3(5):493–530, 1989. (Also IlliGAL Report No. 89003).

[15] D. E. Goldberg and P. Segrest. Finite Markov chain analysis of genetic algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, pages 1–8, 1987.

[16] P. Hansen, N. Mladenovic, and D. Perez-Britos. Variable neighborhood decomposition search. *Journal of Heuristics*, 7(4):335–350, 2001.

[17] G. Harik, E. Cantú-Paz, D. E. Goldberg, and B. L. Miller. The gambler's ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, 7(3):231–253, 1999. (Also IlliGAL Report No. 96004).

[18] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[19] T. Jansen and I. Wegener. On the analysis of evolutionary algorithms - a proof that crossover really can help. *Algorithmica*, 34(1):47–66, 2002.

[20] T. Jansen and I. Wegener. Real royal road functions - where crossover provably is essential. *Discrete Applied Mathematics*, pages 111–125, 2005.

[21] M. Kimura. Diffusion models in population genetics. *Journal of Applied Probability*, 1:177–232, 1964.

[22] C. F. Lima, M. Pelikan, K. Sastry, M. V. Butz, D. E. Goldberg, and F. Lobo. Substructural neighborhoods for local search in the bayesian optimization algorithm. *Parallel Problem Solving from Nature (PPSN IX)*, pages 232–241, 2006. (Also IlliGAL Report No. 20060).

[23] S. Luke. When short runs beat long runs. *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 74–80, 2001.

[24] B. L. Miller. *Noise, Sampling, and Efficient Genetic Algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, May 1997. (Also IlliGAL Report No. 97001).

[25] B. L. Miller and D. E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9(3):193–212, 1995. (Also IlliGAL Report No. 95006).

[26] M. Mitchell, J. Holland, and S. Forrest. When will a genetic algorithm outperform hill-climbing. *Advances in Nueral Information Processing Systems*, 6:51–58, 1994.

[27] H. Mühlenbein. Evolution in time and space- the parallel genetic algorithm. *Foundations of Genetic Algorithms*, pages 316–337, 1991.

[28] H. Mühlenbein. How genetic algorithms really work: Mutation and hillclimbing. *Parallel Problem Solving from Nature II*, pages 15–26, 1992.

[29] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong. Classification of adaptive memetic algorithms: A comparative study. *IEEE Transactions On Systems, Man and Cybernetics - Part B*, 36(1):141–152, 2006.

[30] W. M. Rudnick. *Genetic algorithms and fitness variances with an application to the automated design of artificial neural networks*. PhD thesis, Oregon Graduate Institute of Science & Technology, Beaverton, OR, 1992.

[31] K. Sastry. Evaluation-relaxation schemes for genetic and evolutionary algorithms. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2001. (Also IlliGAL Report No. 2002004).

[32] K. Sastry and D. E. Goldberg. Designing competent mutation operators via probabilistic model building of neighborhoods. *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, 2:114–125, 2004. Also IlliGAL Report No. 2004006.

[33] K. Sastry and D. E. Goldberg. Let's get ready to rumble: Crossover versus mutation head to head. *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, 2:126–137, 2004. Also IlliGAL Report No. 2004005.

[34] K. Sastry and D. E. Goldberg. Let's get ready to rumble redux: Crossover versus mutation head to head on exponentially scaled problems. IlliGAL Report No. 2007005, University of Illinois at Urbana-Champaign, Urbana, IL, February 2007.

[35] K. Sastry, P. Winward, D. E. Goldberg, and C. F. Lima. Fluctuating crosstalk as a source of deterministic noise and its effects on ga scalability. *Applications of Evolutionary Computing EvoWorkshops2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoInteraction, EvoMUSART, EvoSTOCK*, pages 740–751, 2006. (Also IlliGAL Report No. 2005025).

[36] R. Srivastava. Time continutation in genetic algorithms. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002.

[37] R. Srivastava and D. E. Goldberg. Verification of the theory of genetic and evolutionary continuation. *Proceedings of the 2001 Genetic and Evolutionary Computation Conference*, pages 551–558, 2001. (Also IlliGAL Report No. 2001007).

[38] T. Storch and I. Wegener. Real royal road functions for constant population size. *Theoretical Computer Science*, 320:123–134, 2004.

[39] D. Thierens and D. E. Goldberg. Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving from Nature*, 3:116–121, 1994.

[40] D. Thierens, D. E. Goldberg, and A. G. Pereira. Domino convergence, drift, and the temporal-salience structure of problems. *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 535–540, 1998.

[41] J.-P. Watson, A. Howe, and L. Whitley. An analysis of iterated local search for job-shop scheduling. *Proceedings of the Fifth Metaheuristics International Conference*, pages 1101–1106, 2003.