

# Empirical Analysis of Ideal Recombination on Random Decomposable Problems

Kumara Sastry<sup>1</sup>, Martin Pelikan<sup>2</sup>, and David E. Goldberg<sup>1</sup>

<sup>1</sup> Illinois Genetic Algorithms Laboratory (IlligAL), Industrial and Enterprise Systems Engineering  
University of Illinois at Urbana-Champaign, Urbana IL 61801

<sup>2</sup>Department of Math and Computer Science, University of Missouri at St. Louis. St. Louis MO 63121  
ksastry@uiuc.edu, deg@uiuc.edu, pelikan@cs.umsl.edu

## ABSTRACT

This paper analyzes the behavior of a selectorecombinative genetic algorithm (GA) with an *ideal* crossover on a class of random additively decomposable problems (rADPs). Specifically, additively decomposable problems of order  $k$  whose subsolution fitnesses are sampled from the standard uniform distribution  $U[0, 1]$  are analyzed. The scalability of the selectorecombinative GA is investigated for 10,000 rADP instances. The validity of facetwise models in bounding the population size, run duration, and the number of function evaluations required to successfully solve the problems is also verified. Finally, rADP instances that are easiest and most difficult are also investigated.

## Categories and Subject Descriptors

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

## General Terms

Algorithms, Experimentation, Performance

## Keywords

Genetic algorithms, ideal crossover, test problems, additively-decomposable problems, empirical analysis, building blocks, population sizing, convergence time, scalability analysis, problem difficulty

## 1. INTRODUCTION

In the last two decades, a class of competent genetic algorithms (GAs)—GAs that solve boundedly difficult problems quickly, reliably, and accurately—have been developed and their scalability has been tested on a class of problems that are adversarial in nature [13]. Facetwise mod-

els have also been developed to bound the scalability of selectorecombinative GAs, which show that GAs that accurately identify and effectively exchange key substructures of additively-decomposable problems, scale polynomially (usually sub-quadratically) with the problem size.

In this paper, we consider a broader class of additively decomposable problems and analyze the behavior of selectorecombinative GAs. Specifically, we analyze a class of additively decomposable problems, where the fitnesses of competing subsolutions within a partition are sampled from the standard uniform distribution. Moreover, in the spirit of facetwise and dimensional thinking, we consider a selectorecombinative GA with an *ideal* recombination operator [13]. The *ideal* crossover assumes that the substructures of the underlying search problems are known and exchanges subsolutions without disrupting them. Additionally, the use of the perfect crossover eliminates any effects of inaccuracies in the identification of important substructures and permits us to investigate the performance of selectorecombinative GAs in what can be considered a best-case scenario. The results confirm the validity and applicability of facetwise models in bounding the population size, run duration, and the number of function evaluations required to successfully solve the rADP instances.

This paper is organized as follows. We provide a brief background on GA problem difficulty and test problem design in the section 2, followed by an outline of facetwise models for bounding the scalability of selectorecombinative GAs in section 3. We introduce the class of rADPs in section 4 followed by their analysis in section 5. Finally, we provide summary and key conclusions in section 6.

## 2. ADVERSARIAL APPROACH FOR TEST PROBLEM DESIGN

A powerful and common approach used in designing complex systems is to test them on the boundary of their *design envelope*. For example, an airplane is designed to fly a range of missions under a variety of externally imposed conditions. The combination of missions and external conditions determines a *performance envelope* in which we expect the aircraft to perform well. As long as we operate within the specified envelope, and as long as the aircraft has been properly designed, we expect the aircraft to perform satisfactorily.

Elsewhere Goldberg [13] argues that taking a similar *design envelope* point of view benefits the design and analysis of scalable genetic and evolutionary algorithms as well. By

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO '07, July 7–11, 2007, London, England, United Kingdom.  
Copyright 2007 ACM 978-1-59593-697-4/07/0007 ...\$5.00.

considering a *design envelope* point of view, he advocates designing test problems characterized by various *dimensions of problem difficulty* that thwart the search mechanism of selectorecombinative GAs in the extreme [13]. That is, to consider bounding *adversarial problems* that exploit one or more dimensions of problem difficulty. One dimension of problem difficulty is *intra-building-block* difficulty, where the difficulty is generated from *within* a substructure [13], and *deceptive* problems belong to this class of difficulty.

One such class of *adversarial* problems is the *m-k* deceptive *trap* problem, which consists of additively separable *deceptive* functions [8, 1, 12, 20, 29, 10]. Deceptive functions are designed to thwart the very mechanism of selectorecombinative search by punishing any localized hillclimbing and requiring mixing of whole building blocks (BBs) at or above the order of deception. Using such *adversarially* designed functions is a stiff test—in some sense the stiffest test—of algorithm performance. The idea is that if an algorithm can beat an adversarially designed test function, it can solve other problems that are equally hard or easier than the adversarial problem. Here, we verify if this assertion holds for a class of random additively decomposable problems.

### 3. SCALABILITY OF SELECTORECOMBINATIVE GENETIC ALGORITHMS

Two key factors for predicting the scalability of a selectorecombinative GA are the convergence time and population sizing. Therefore, we present facetwise models of convergence time and population sizing. More details and motivation for using facetwise modeling and dimensional analysis are given elsewhere [13].

#### 3.1 Population-Sizing Model

Goldberg, Deb and Clark [14] proposed population-sizing models for correctly deciding between competing BBs. They incorporated noise arising from other partitions into their model. Harik et al [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 the best and second best BBs in a partition as a gambler’s ruin problem [11]. Here we use an approximate form of the population-sizing model proposed by Harik et al [17]:

$$n = \frac{\sqrt{\pi} \sigma_{BB}}{2d} 2^k \sqrt{m} \log m, \quad (1)$$

where  $k$  is the BB size,  $m$  is the number of BBs,  $d$  is the signal between the competing BBs, and  $\sigma_{BB}$  is the fitness variance of a building block. The above equation assumes a failure probability,  $\alpha = 1/m$ .

#### 3.2 Convergence-Time Model

Mühlenbein and Schlierkamp-Voosen [22] derived a convergence-time model for the breeder GA using the notion of *selection intensity* from population genetics [9]. Selection-intensity based models have since been developed for other selection schemes and for noisy environments [28, 21, 3]. Even though the selection-intensity-based convergence-time models are derived for the OneMax problem, they are generally applicable to additively decomposable problems as well:

$$t_c = \frac{c_t}{I} \sqrt{m}, \quad (2)$$

where  $I$  is the selection intensity, and  $c_t = 4.72$  is an empirically-determined constant. For binary tournament selection,  $I = 1/\sqrt{\pi}$ .

The above model yields good characterization of population convergence when even from early on in the GA run, only two subsolutions compete with each other. However, when the best subsolution competes with more than one subsolution, stochastic errors in finite population can accumulate, resulting in significant fluctuations in the proportions of competing subsolutions due to *genetic drift* [19, 16, 2]. The *drift time* required for convergence to the optimal subsolution is given by [16],

$$t_d = 6p_0(1 - p_0)n. \quad (3)$$

Assuming a worst-case scenario where all  $2^k - 1$  subsolutions compete with the optimal subsolution, that is setting  $p_0 = 1/2^k$ , and substituting Equation 1 for  $n$  in the above equation we get

$$t_d = 3\sqrt{\pi} \cdot \frac{\sigma_{BB}}{d} \cdot \sqrt{m} \log m. \quad (4)$$

Using equations 1 and 2, we can now predict the scalability, or the number of function evaluations required for successful convergence, of GAs as follows:

$$n_{fe} = n \cdot t_c = c_f \frac{\sigma_{BB}}{d} \cdot 2^k \cdot m \log m, \quad (5)$$

where  $c_f = c_t \sqrt{\pi}/2I$ .

### 4. RANDOM ADDITIVELY DECOMPOSABLE PROBLEMS (RADPS)

In this section, we describe a class of random additively decomposable problem. Here, we only consider problems without overlap between substructures, which is considered elsewhere [23]. We assume that the candidate solutions are represented by binary strings of fixed length. Here, we restrict our analysis to boundedly-difficult problems of order  $k$ .

An  $\ell$ -bit additively decomposable problem of order  $k$  consists of  $m$  subproblems of  $k$  bits each. When there is no overlap between the subproblems, each bit belongs to exactly one of the subproblems. Therefore, the total number of bits is  $\ell = m \cdot k$ . The fitness function for an additively decomposable problem can be written as

$$\mathcal{F}(X_1, X_2, \dots, X_\ell) = \sum_{i=1}^m g_i(S_i) \quad (6)$$

where  $m$  is the number of subproblems,  $g_i$  is the fitness of the  $i^{\text{th}}$  subproblem, and  $S_i \subset X_1, X_2, \dots, X_\ell$  is the subset of variables of order  $k$  for the  $i^{\text{th}}$  subproblem.

In order to ease the analysis, in this study we consider the subproblems to be identical, that is,  $g_1 = g_2 = \dots = g_m$ . We note that this restriction can be relaxed and the behavior of selectorecombinative GAs is qualitatively similar [23]. We further assume that the subproblem is generated randomly from the standard uniform distribution  $U[0, 1]$ . That is, the fitness contributions of  $2^k$  subsolutions are sampled from the standard uniform distribution  $U[0, 1]$ . The same fitness contribution is used to compute the fitness of subsolutions from all the partitions.

To estimate the population size and the drift time for rADPs, we need to estimate the noise-to-signal ratio. To

do so, we estimate the probability density function (p.d.f.) of the signal difference  $d$  and the building-block variance (noise)  $\sigma_{BB}$ .

The signal difference  $d$  is defined as the fitness difference between the best and the second-best subsolution. Since the subsolution fitness is sampled from a uniform distribution, the  $k^{\text{th}}$  order statistic follows a Beta distribution with parameters  $\alpha = k$  and  $\beta = 2^k - k + 1$  [5, 4, 7, 6, 18]. From order statistics, we know that if  $X$  has a p.d.f.  $f(x)$  and a cumulative density function  $F(x)$ , then the probability distribution of the  $r^{\text{th}}$  order statistic,  $f_{U_{r:N}}$  is given by

$$f_{U_{r:2^k}} = \frac{2^k!}{(r-1)!(2^k-r)!} \cdot [F(x)]^{r-1} [1-F(x)]^{2^k-r} f(x), \quad (7)$$

for  $r = 1, 2, \dots, 2^k$ , with  $r = 1$  and  $r = 2^k$  denoting the index of the worst and best subsolutions, respectively. Substituting the p.d.f. and cumulative density function of the standard uniform distribution,  $f(x) = 1$ , and  $F(x) = x$ , respectively, in the above equation we get,

$$f_{U_{r:2^k}} = \frac{2^k!}{(r-1)!(2^k-r)!} x^{r-1} (1-x)^{2^k-r}. \quad (8)$$

Therefore, the p.d.f of the fitness of the best subsolution  $f_{U_{2^k:2^k}}$  is given by

$$\begin{aligned} f_{U_{2^k:2^k}}(x) &= \frac{2^k!}{(2^k-1)!0!} x^{2^k-1} (1-x)^0, \\ &= 2^k x^{2^k-1}. \end{aligned} \quad (9)$$

Similarly, the p.d.f of the fitness of the second-best subsolution  $f_{U_{2^k-1:2^k}}(y)$  can be written as

$$\begin{aligned} f_{U_{2^k-1:2^k}}(y) &= \frac{2^k!}{(2^k-2)!1!} x^{2^k-2} (1-x)^1, \\ &= 2^k (2^k-1) \cdot y^{2^k-2} (1-y). \end{aligned} \quad (10)$$

To summarize, the p.d.fs of the fitness of the best subsolution  $f_{U_{2^k:2^k}}(x)$  and the  $2^{\text{nd}}$ -best subsolution  $f_{U_{2^k-1:2^k}}(y)$  is given by

$$\begin{aligned} f_{U_{2^k:2^k}}(x) &= \text{Beta}(2^k, 1) = 2^k \cdot x^{2^k-1}, \quad (11) \\ f_{U_{2^k-1:2^k}}(y) &= \text{Beta}(2^k-1, 2), \\ &= 2^k (2^k-1) \cdot y^{2^k-2} (1-y). \end{aligned} \quad (12)$$

To compute the signal, we first need to compute the joint p.d.f of the best and the second best subsolutions. From order statistics, we know that the joint p.d.f of  $i^{\text{th}}$  and  $j^{\text{th}}$  order statistics is

$$\begin{aligned} g_{U_{i:N}, U_{j:N}}(y, x) &= \frac{N!}{(i-1)!(j-i-1)!(N-j)!} \cdot \\ &\quad [F(y)]^{i-1} [F(x)-F(y)]^{j-i-1} \\ &\quad [1-F(x)]^{N-j} f(x)f(y), \end{aligned} \quad (13)$$

where  $y < x$ , and  $i < j$ . Since both the fitness of best and second-best subsolutions,  $X$  and  $Y$ , are sampled from the standard uniform distribution,  $f(x) = 1$ ,  $f(y) = 1$ ,  $F(x) = x$ , and  $F(y) = y$ . Additionally,  $i = 2^k - 1$ ,  $j = 2^k$ , and  $N = 2^k$ . Substituting these values in the above equation we

can write the joint p.d.f of the fitnesses of best and  $2^{\text{nd}}$ -best subsolutions as

$$\begin{aligned} g_{U_{i:N}, U_{j:N}}(y, x) &= \frac{2^k!}{(2^k-2)!0!0!} \cdot y^{2^k-2} (x-y)^0 (1-x)^0, \\ &= 2^k (2^k-1) y^{2^k-2} \end{aligned} \quad (14)$$

Since, the signal difference  $d = x - y$ , the p.d.f. of  $d$  is given by

$$\begin{aligned} f_d(d) &= \int_0^1 \int_0^{1-d} g_{U_{i:N}, U_{j:N}}(y, x) dy \cdot dx \\ &= 2^k (1-d)^{2^k-1}. \end{aligned} \quad (15)$$

We know that for the standard uniform distribution,  $\mu_2 = 1/12$  and  $\mu_4 = 1/80$ . Therefore, the mean and the variance of the subsolution variance is given by  $\frac{(2^k-1)}{2^k} \mu_2 \approx \mu_2 = 1/12$ , and

$$\begin{aligned} E[(\sigma_{BB}^2 - E[\sigma_{BB}^2])^2] &= \frac{(\kappa-1)}{\kappa^3} [(\kappa-1)\mu_4 - (\kappa-3)\mu_2^2], \\ &\approx \frac{0.0056}{2^k}, \end{aligned} \quad (16)$$

respectively, where  $\kappa = 2^k$ . We can approximate the p.d.f. of the variance of subsolution fitnesses as the normal distribution with parameters  $\mu = \sigma_U^2 = 1/12$ , and  $\sigma^2 = 0.0056/2^k$ :

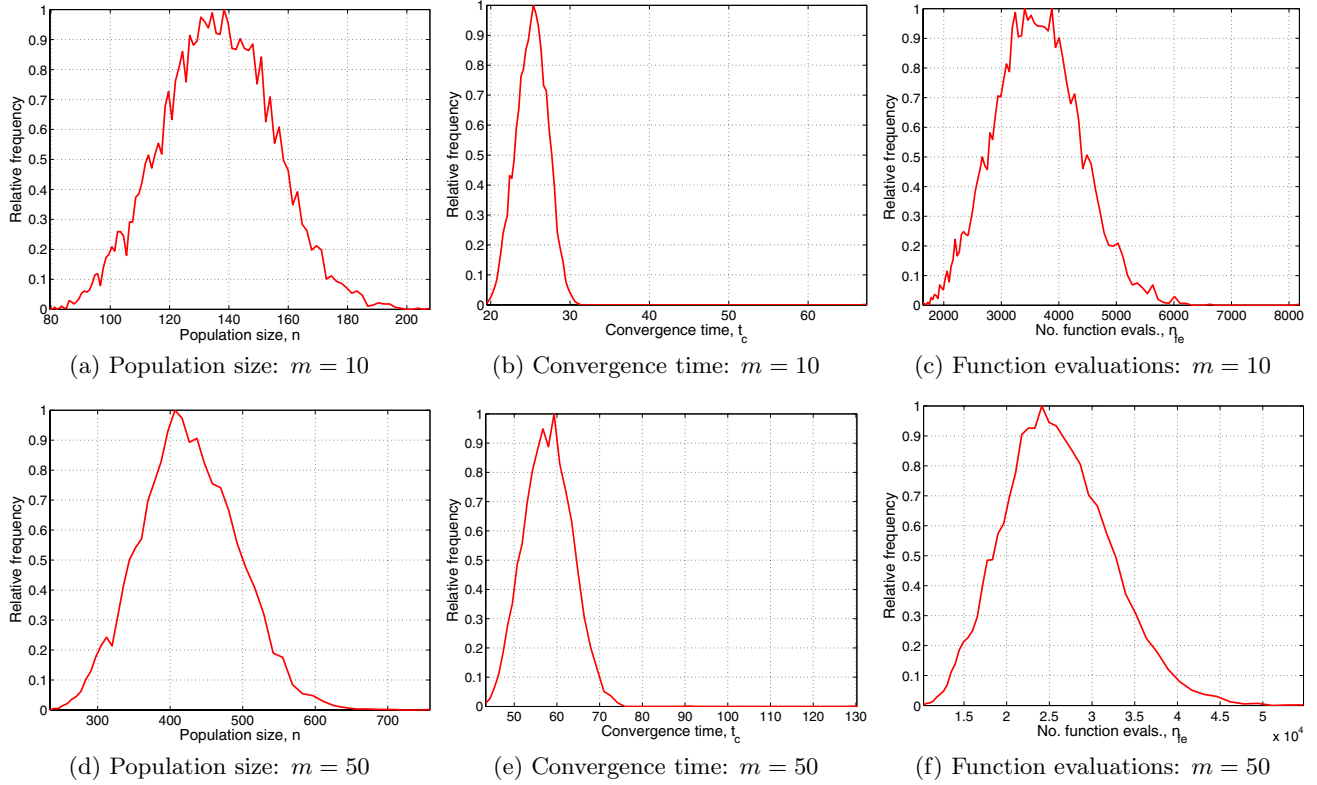
$$f_{\sigma_{BB}^2}(\sigma_{BB}^2) \sim \mathcal{N}\left(\frac{1}{12}, \frac{0.075}{\sqrt{2^k}}\right). \quad (17)$$

From the p.d.fs of  $d$  and  $\sigma_{BB}^2$ , we can easily see that  $E[1/d] = 2^k$  and  $E[\sigma_{BB}^2] = 1/12$ . Using these values in Equations 1, 4, and 5, we can bound the scalability of the GA with *ideal* crossover for rADPs as discussed in the following section.

## 5. ANALYSIS OF RADPS

We generated 10,000 rADP instances for our analysis and considered problems with  $m = 5$  to  $m = 50$  subproblems of order  $k = 3, 4$ , and  $5$ . For brevity, we only show results for  $k = 4$  in this paper as the results are qualitatively similar for  $k = 3$  and  $5$ . We considered a generationwise selectorecombinative GA with non-overlapping populations of fixed size. We use a binary tournament selection without replacement [15, 25], and a uniform BB-wise crossover [26]. In uniform BB-wise crossover, two parents are randomly selected from the mating pool and their subsolutions in each partition are exchanged with a probability of 0.5. Therefore, none of the subsolutions are disrupted during a recombination event. The GA run is terminated when all the individuals in the population are identical. The average number of BBs correctly converged are computed over 50 independent runs. The minimum population size required such that  $m-1$  BBs converge to the correct value is determined by a bisection method [24]. The results of population-size is averaged over 30 such bisection runs, while the results for  $t_c$  and  $n_{fe}$  are averaged over 1,500 independent GA runs.

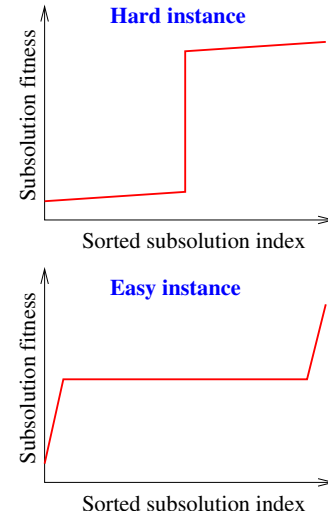
We begin our analysis by comparing the histogram of the population size, run duration, and number of function evaluations for a given  $m$  in Figure 1. Specifically we plot the relative frequency—where a particular frequency is divided by the maximum frequency—of the rADPs that require a



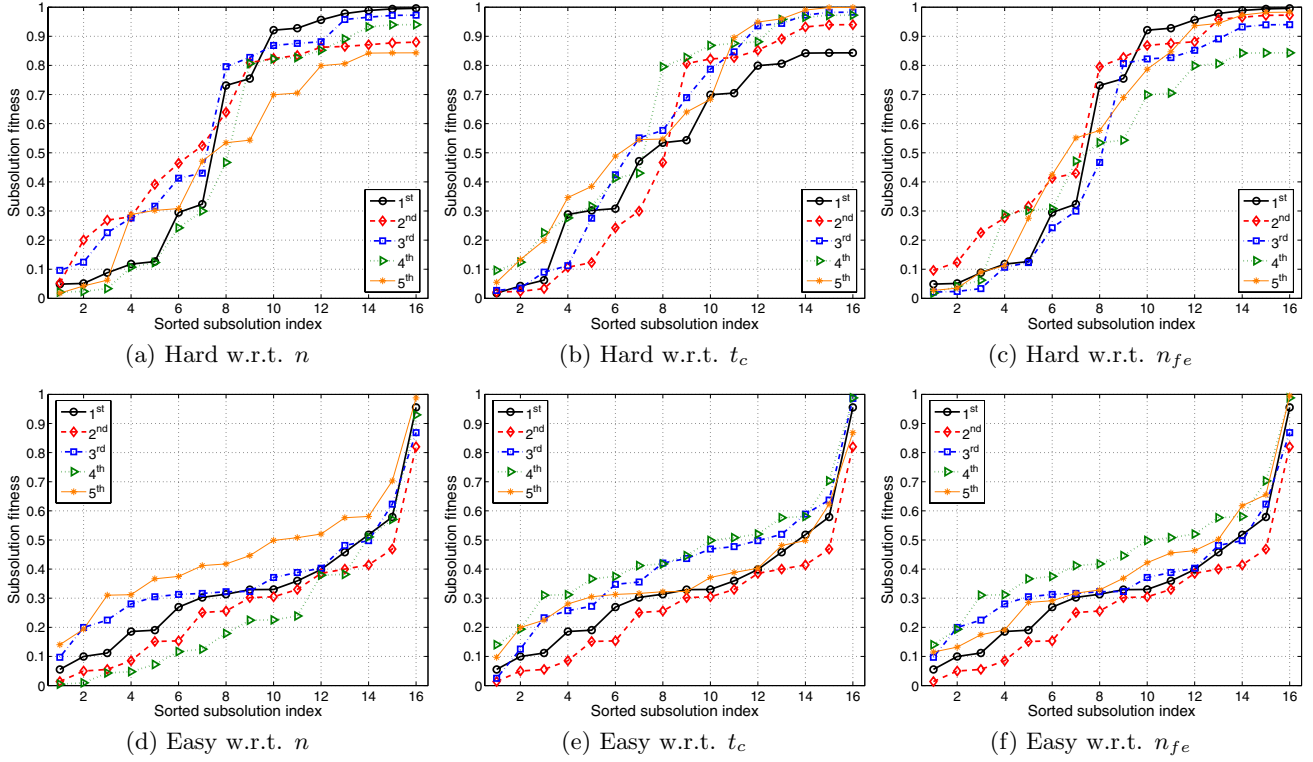
**Figure 1: Histogram of population size, convergence time in terms of number of generations, and the number of function evaluations required to successfully solve at least  $m - 1$  out of  $m$  building blocks correctly for  $m = 10$  and  $m = 50$ . To obtain the relative frequency, all frequencies have been divided by the maximum frequency. The plots show the results for 10,000 rADPs and are averaged over 30 bisection runs with 50 independent GA runs for each bisection trial.**

certain population, run-duration, and number of function evaluations ranges to solve at least  $m - 1$  out of  $m$  sub-problems to optimality for  $m = 10$ , and 50. The results for other problem sizes are qualitatively similar and are provided elsewhere [27]. The results show that the histograms have log-normal behavior with the tail increasing with  $m$ , which indicates the presence of drift in solving some rADPs. From the histograms we find that 0.08–0.3%, 0.01–0.1%, and 0.15–0.59% of the rADP instances require  $n$ ,  $t_c$ , and  $n_{fe}$  greater than three standard deviations from the median, respectively.

Next, we investigate easy and hard rADP instances according to population-size, convergence-time, and function-evaluation requirements as the function of number of sub-solutions. Based on facetwise analysis we expect the hard instances to have minimum signal-to-noise ratio and the easy instances to have maximum signal-to-noise ratio. That is, for hard instances we expect to have minimum signal and maximum variance. Ideally, for hard instances, about half the sub-solutions to have low fitness values and the other half to have high fitness values (see Figure 2). Thus, not only the variance is maximized, but also the signal is close to zero. For the easy instances, we expect to have maximum signal and minimum variance. Therefore, as shown in Figure 2, we expect the best sub-solution to have significantly higher fitness than other sub-solutions and the rest to have near-equal fitness.



**Figure 2: Ideal easy and hard problem instances based on facetwise theory.**



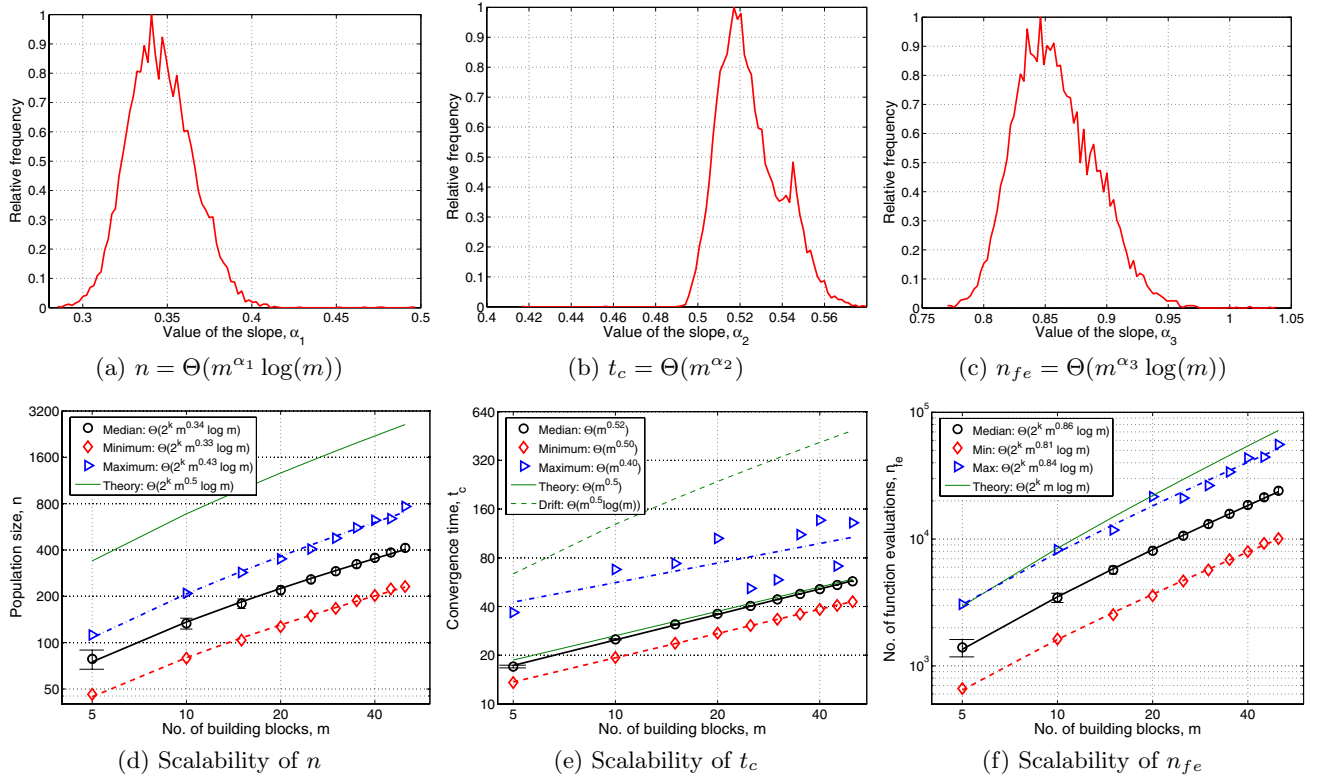
**Figure 3:** The top five difficult and easy rADP instances with respect to the population size  $n$ , run duration  $t_c$ , and the number of function evaluations  $n_{fe}$ . The subsolutions are sorted according to their fitness values. The 10,000 rADPs are ranked in descending according to  $n$ ,  $t_c$ , and  $n_{fe}$  values for a given  $m$  and they are averaged over all  $m$  values (5–50), and the top 5 are depicted as the difficult instances and the bottom 5 are depicted as the easy instances.

We show empirically observed easy and hard rADP instances according to population-size, convergence-time, and function-evaluation requirements for  $m = 5-50$ . Specifically, for each  $m$ , we rank each rADP in descending order of the population size required. Then for each rADP, we sum up its ranks over all  $m$  values and consider this as the overall rank of the rADP instance. We then select the top 5 and bottom 5 rADP instances according to the overall rank. We repeat the same procedure for  $t_c$  and  $n_{fe}$  and show the five hardest—in terms of requiring maximum  $n$ ,  $t_c$ , or  $n_{fe}$  over all values of  $m$ —and five easiest—in terms of requiring minimum  $n$ ,  $t_c$ , or  $n_{fe}$  over all values of  $m$ —rADP instances in Figure 3. The results qualitatively validate our expectation and clearly show that for hard rADP instances, there are more subsolutions competing with the best subsolution. Moreover, the difference in the fitness of competing subsolutions is very small, that is, hard problem instances have low signal  $d$ . Additionally, since there are more than two competing subsolutions in each partition, we can expect drift to play a major role as the given collateral noise [14]. In contrast, easy rADP instances all have two competing subsolutions with a large signal. These results agree with the models of GA-design theory [13].

One of our primary interests is to investigate how the population size, convergence time, and the number of function evaluations scale with the problem size. From the facetwise models (Equations 1, 2, and 5), we expect  $n$ ,  $t_c$ , and  $n_{fe}$  to scale as  $\Theta(m^{\alpha_1} \log m)$ ,  $\Theta(m^{\alpha_2})$ , and  $\Theta(m^{\alpha_3} \log m)$ , res-

spectively. We begin by plotting the histograms  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  in Figures 4(a)–(c). Specifically, we run each of the 10,000 rADP instance with increasing number of building blocks and analyze the scalability of population size, convergence time, and number of function evaluations as a function of problem size. Then using a least-squares method, we estimate the coefficient values of scalability for each rADP instance. The histograms represent the different values for  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  estimated for different rADP instances. The results show that the population size required to solve at least  $m - 1$  subproblems to optimality scales on an average as  $\Theta(m^{0.35} \log m)$  and in the worst case scales as  $\Theta(m^{0.5} \log m)$  as predicted by the facetwise model (Equation 1). Similarly, on an average  $t_c$  and  $n_{fe}$  scale as  $\Theta(m^{0.52})$  and  $\Theta(m^{0.85} \log m)$ , and in the worst case they scale  $\Theta(m^{0.58})$  and  $\Theta(m^{1.04} \log m)$  respectively, agreeing with facetwise models (Equation 2, 3, and 5).

We also plot the median, minimum, and maximum  $n$ ,  $t_c$ , and  $n_{fe}$  required over the 10,000 rADP instances and compare it to the facetwise models in Figures 4(d)–(f). The results show that the facetwise models clearly bound the scalability of selectorecombinative GAs on rADPs. From Figure 4(e), we can see that while the convergence-time model (Equation 2) bounds the empirically observed  $t_c$  on an average, the drift-time model (Equation 3) yields an upper bound on the run duration. More importantly, the results clearly validate the assertion that testing and modeling selectorecombinative GAs on adversarially-designed test prob-



**Figure 4: Histogram of the scalability of population size, convergence time, and number of function evaluations with problem size for 10,000 rADPs. From the facetwise models, we use  $n = \Theta(m^{\alpha_1} \log m)$ , and  $n_{fe} = \Theta(m^{\alpha_3} \log m)$ . We also show the scalability of the median, minimum, and maximum values of  $n$ ,  $t_c$ , and  $n_{fe}$  with  $m$ .**

lems bounds their performance on other problems that are equally hard or easier than the adversarial function for the case of rADPs.

Finally, we analyze the easy and hard rADP instances in terms of scalability and the results are shown in Figure 5. Unlike the problems shown in Figure 3, here we show five rADP instances that are easy and hard in terms of scalability of  $n$ ,  $t_c$ , and  $n_{fe}$ . The results show that the problems that are most difficult in terms of scalability of  $n$  are rADP instances where there are two competing subsolutions with near-equal fitness (very low signal  $d$ , or a very high noise-to-signal ratio). It is interesting to note that the population-sizing model was derived based on this very scenario of two competing subsolutions. Similarly the most difficult in terms of scalability of  $t_c$  are rADP instances with more than two competing subsolutions with near-equal fitness, where we can expect drift to play a dominating role. Similar to the results shown in Figure 3, the easy rADP instances in terms of scalability also have two competing subsolutions with a reasonably high signal (low noise-to-signal ratio).

## 6. SUMMARY AND CONCLUSIONS

We analyzed the behavior of a selectorecombinative GA with an *ideal* crossover operator on a class of boundedly-difficult random additively decomposable problems (rADPs). Specifically we considered additively decomposable problems of order  $k$ , where the subsolutions fitnesses are sampled from the standard uniform distribution  $U[0, 1]$ .

We verified the validity of facetwise models developed based on adversarial test functions on modeling the behavior of the GA on rADPs. The results show that for each of the 10,000 rADP instances that were generated and tested, the selectorecombinative GA scales sub-quadratically as predicted by the facetwise models. We also analyzed the easy and hard rADP instances, which show that easy rADP instances have two competing subsolutions with significant difference in fitness, while hard rADP instances have two or more competing subsolutions with near-equal fitness.

This study verifies the validity and applicability of facetwise GA scalability theory on a broad class of problems as opposed to a few specific adversarial problems. This study also validates the *adversarial* approach towards designing test problems that test the GA designs on the *design envelope*. Finally, the results provides empirical evidence that testing GAs on adversarial problems of bounded difficulty provides a useful bound for sub-class of difficult decomposable problems whose difficulty is less than that of the adversarial problem.

## Acknowledgments

This work was 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 NSF CAREER grant ECS-0547013 and ITR grant DMR-03-25939 at Materials Computation Center, UIUC. The work was also supported by the Research Award and the Research

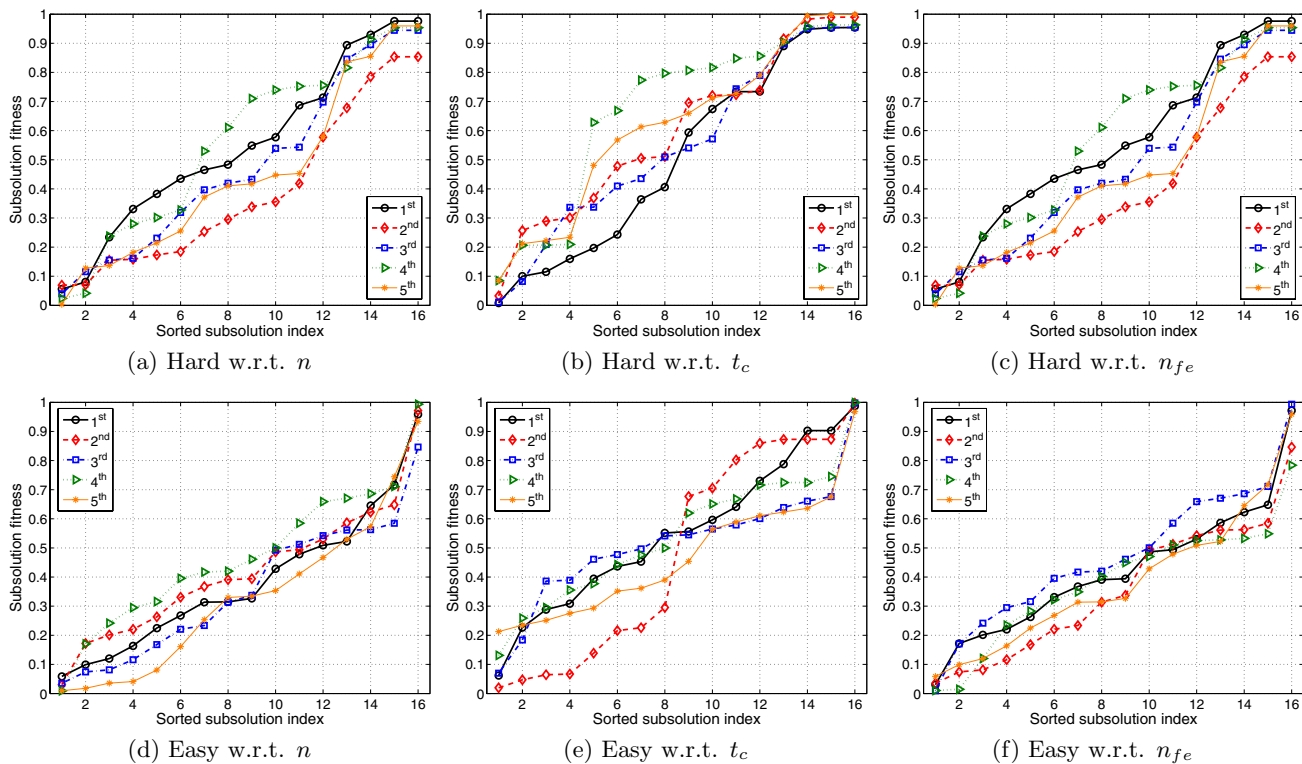


Figure 5: The top five difficult and easy rADP instances with respect to the scalability of population size  $n$ , run duration  $t_c$ , and the number of function evaluations  $n_{fe}$ .

Board at the University of Missouri. 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] D. H. Ackley. *A connectionist machine for genetic hill climbing*. Kluwer Academic Publishers, 1987.
- [2] 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.
- [3] T. Bäck. Generalized convergence models for tournament—and  $(\mu, \lambda)$ —selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 2–8, 1995.
- [4] N. Balakrishnan and W. W. S. Chen. *Handbook of tables for order statistics from lognormal distributions with applications*. Kluwer Academic Publishers, Amsterdam, Netherlands, 1999.
- [5] N. Balakrishnan and A. C. Cohen. *Order statistics and inference*. Academic Press, New York, NY, 1991.
- [6] N. Balakrishnan and C. R. Rao, editors. *Order statistics: Applications*. Elsevier, Amsterdam, Netherlands, 1999.
- [7] N. Balakrishnan and C. R. Rao, editors. *Order statistics: Theory and methods*. Elsevier, Amsterdam, Netherlands, 1999.
- [8] A. D. Bethke. *Genetic algorithms as function optimizers*. PhD thesis, University of Michigan, Ann Arbor, MI, 1981. (University Microfilms No. 8106101).
- [9] M. G. Bulmer. *The Mathematical Theory of Quantitative Genetics*. Oxford University Press, Oxford, 1985.
- [10] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms*, 2:93–108, 1992. (Also IlliGAL Report No. 91009).
- [11] W. Feller. *An Introduction to Probability Theory and its Applications*. Wiley, New York, NY, 1970.
- [12] D. E. Goldberg. Simple genetic algorithms and the minimal deceptive problem. In L. Davis, editor, *Genetic algorithms and simulated annealing*, chapter 6, pages 74–88. Morgan Kaufmann, Los Altos, CA, 1987.
- [13] D. E. Goldberg. *Design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.
- [14] 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).

- [15] 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).
- [16] 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.
- [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] R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Macmillan, New York, NY, 5th edition, 1995.
- [19] M. Kimura. Diffusion models in population genetics. *Journal of Applied Probability*, 1:177–232, 1964.
- [20] G. E. Liepins and M. D. Vose. Representational issues in genetic optimization. *Journal of Experimental and Theoretical Artificial Intelligence*, 2:101–115, 1990.
- [21] 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).
- [22] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. continuous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.
- [23] M. Pelikan, K. Sastry, M. V. Butz, and D. E. Goldberg. Hierarchical BOA on random decomposable problems. IlliGAL Report No. 2006002, University of Illinois at Urbana Champaign, Urbana, IL, January 2006.
- [24] 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).
- [25] K. Sastry and D. E. Goldberg. Modeling tournament selection with replacement using apparent added noise. *Intelligent Engineering Systems Through Artificial Neural Networks*, 11:129–134, 2001. (Also IlliGAL Report No. 2001014).
- [26] 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.
- [27] K. Sastry, M. Pelikan, and D. E. Goldberg. Analysis of ideal recombination on random decomposable problems. IlliGAL Report No. 2006016, University of Illinois at Urbana-Champaign, Urbana, IL, April 2006.
- [28] D. Thierens and D. E. Goldberg. Convergence models of genetic algorithm selection schemes. *Parallel Problem Solving from Nature*, 3:116–121, 1994.
- [29] L. D. Whitley. Fundamental principles of deception in genetic search. *Foundations of Genetic Algorithms*, pages 221–241, 1991.