Limits of Scalability of Multiobjective Estimation of Distribution Algorithms

Kumara Sastry[1], **Martin Pelikan**[2], **David E. Goldberg**[1]
[1]Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of General Engineering
University of Illinois at Urbana-Champaign, Urbana IL 61801
[2]Department of Mathematics and computer Science
University of Missouri at St. Louis, St. Louis MO 63121
ksastry@uiuc.edu, pelikan@cs.umsl.edu, deg@uiuc.edu

Abstract- The paper analyzes the scalability of multiobjective estimation of distribution algorithms (MOEDAs), particularly multiobjective extended compact genetic algorithm (meCGA), on a class of boundedly-difficult additively-separable multiobjective optimization problems. The paper demonstrates that even if the linkage is correctly identified, massive multimodality of the search problems can easily overwhelm the nicher and lead to exponential scale-up. The exponential growth of the Pareto-optimal solutions introduces a fundamental limit on the scalability of MOEDAs and the number of competing sub-structures between the multiple objectives. Facetwise models are subsequently used to predict this limit in the growth rate of the number of differing substructures between the two objectives to avoid the niching method from being overwhelmed and lead to polynomial scalability of MOEDAs.

1 Introduction

One of the challenging areas in genetic and evolutionary computation that has received increased attention is multiobjective evolutionary algorithms (MOEAs). Several MOEAs have been proposed and applied with significant success to real-world problems [1, 2]. However, studies on the theory and analysis of MOEAs have been limited in part because of the complexity of both the algorithms and the problems. For example, some aspects of problem difficulty and algorithm scalability have been recently studied [3, 4].

Recently, there has been a growing interest in extending estimation of distribution algorithms [5, 6]—a class of *competent* genetic algorithms [7] that replace traditional variation operators of genetic algorithms (GAs) with probabilistic model building of promising solutions and sampling the model to generate new offspring—to solve multiobjective search problems quickly, reliably, and accurately. Such multiobjective EDAs (MOEDAs) [8, 9, 10, 11] typically combine the model-building and sampling procedures of EDAs with the selection procedure of MOEAs such as the non-dominated sorting GA (NSGA-II) [12], and a niching method such as sharing or crowding in objective space. MOEDAs have been shown to outperform traditional MOEAs in efficiently searching and maintaining Pareto-optimal solutions on boundedly-difficult problems.

However, the scalability of the population size and the number of function evaluations required by EDAs as a function of problem size and the number of Pareto-optimal solutions has been largely ignored. This is the case even though one of the primary motives for designing MOEDAs is to carry over the polynomial (oftentimes sub-quadratic) scalability of EDAs to boundedly-difficulty multiobjective search problems. However, the usual scalability approach used for single-objective EDAs with one or few global solutions—where, we investigate the minimum number of function evaluations to get high-quality solutions quickly, reliably, and accurately—does not work for multiobjective problems, and it is easy to get into combinatorial difficulty. Even with two objectives, additively decomposable problems have exponentially many Pareto-optimal solutions and it is easy to get into combinatorial difficulty. This massive multimodality introduces a fundamental limitation on the scalability of MOEAs in general, and MOEDAs in particular. In this paper, we demonstrate that even if the sub-structures (or linkages) are correctly identified, the combinatorial explosion of the number of Pareto-optimal solutions can overwhelm the niching capability and as expected lead to exponential scalability. This fundamental nature of multiobjective additively decomposable problems introduces a limit on the number of building blocks (or sub-structures) that can differ between the multiple objectives. That is, MOEAs scale polynomially (subquadratically) only if the multiple objectives share common building blocks and have a limited number of building blocks that are different. We use facetwise models to predict the limit in the number of competing sub-structures between multiple objectives.

The paper is organized as follows. We provide a brief background on the motivation for the paper, followed by a brief description of multiobjective extended compact genetic algorithm (meCGA). The details on the test problems and experimental methodologies are described in the subsequent sections. Section 3 demonstrates how massive multimodality of the search problem can overwhelm the niching mechanism and lead to exponential scale-up of MOEDAs. In section 4, using facetwise models of population-sizing for EDAs and niching methods, we propose a method to predict the limit on the growth rate of the number of competing sub-structures between two objectives that can lead to polynomial scalability. Finally, we present key conclusions of the study.

2 How This Paper Came About?

Recently, there has been a growing interest in extending estimation of distribution algorithms (EDAs) to solve multiobjective search problems. Similar to single-objective EDAs [5, 6], multiobjective EDAs (MEDAs) replace the variation operators of MOEAs with the probabilistic model building of promising solutions and sampling the model to generate new offspring. Recently, several MOEDAs have been proposed [8, 9, 10, 13, 14, 15, 11] which have combined variants of the Bayesian optimization algorithm (BOA) [16] and iterated density estimation evolutionary algorithm (IDEA) [17, 18] with the selection and replacement procedures of MOEAs [1, 2]. Even though MEDAs have been shown to outperform their MOEA counterparts on different test problems, none of the studies have systematically analyze the scalability of MOEDAs.

Therefore, our original purpose was to systematically analyze the scalability of MOEDAs on a class of boundedly-difficult additively decomposable problems. We followed a methodology analogous to that used to test the scalability of single-objective EDAs with $\mathcal{O}(1)$ global solutions. In particular, we investigated the minimum number of function evaluations required to obtain and maintain all the Pareto-optimal solutions quickly, reliably, and accurately. Specifically, we investigated the scalability of multiobjective Bayesian optimization algorithm [13] and meCGA on several bi-objective test problems and observed that MOEDAs scale-up exponentially. This is the case even when the EDAs successfully solve each of the objectives alone, requiring only sub-quadratic number of function evaluations.

Further analysis of our scalability results and the test problems, we discovered a fundamental fact of such building-block-wise difficult problems: Exponential growth in the number of Pareto-optimal solutions. When considering each objective in isolation, there is only one global solution, but when considering the two objectives in multiobjective optimization, the total number of global (Pareto-optimal) solutions grow exponentially ($\mathcal{O}(2^m)$). Since we need at least one individual to maintain a Pareto-optimal solution, we need exponentially many individuals to maintain all the Pareto-optimal solutions in the population.

In the following section, we briefly describe multiobjective extended compact genetic algorithm (meCGA), which we use as a representative algorithm of MOEDAs. We choose meCGA not only because of its simplicity and ease of visualizing the probabilistic models, but also because it bounds the scalability of other binary EDAs such as BOA [19, 20].

2.1 Multiobjective Extended Compact Genetic Algorithm (meCGA)

In this study, we use multiobjective extended compact GA and test their scalability on a class of boundedly-difficult problems. The multiobjective extended compact genetic algorithm (meCGA) is similar to mBOA [9], except that the model building and sampling procedure of BOA is replaced with those of extended compact GA (eCGA) [21]. The meCGA is used in this study in part because the simplicity of the probabilistic model and its direct mapping to linkage groups makes it amenable to systematic analysis. The typical steps of meCGA can be outlined as follows:

1. *Initialization:* The population is usually initialized with random individuals. However, other initialization procedures can also be used in a straightforward manner.

2. *Evaluation:* The fitness or the quality-measure of the individuals are computed.

3. *Selection:* As in mBOA, we use the selection procedure of NSGA-II [12]. That is, we first perform the non-dominated sorting, and compute the crowding distance for all the individuals in the population. We then use the individual comparison operator to *bias* the generation of new individuals.

4. *Probabilistic model estimation:* Unlike traditional GAs, however, EDAs assume a particular probabilistic model of the data, or a *class* of allowable models. A *class-selection metric* and a *class-search mechanism* is used to search for an optimum probabilistic model that represents the selected individuals.

   Model representation: The probability distribution used in eCGA is a class of probability models known as marginal product models (MPMs). MPMs partition genes into mutually independent groups and specifies marginal probabilities for each linkage group.

   Class-Selection metric: To distinguish between better model instances from worse ones, eCGA uses a minimum description length (MDL) metric [22]. The key concept behind MDL models is that all things being equal, simpler models are better than more complex ones. The MDL metric used in eCGA is a sum of two components:

   - **Model complexity** which quantifies the model representation size in terms of number of bits required to store all the marginal probabilities:

     $$C_m = \log_2(n) \sum_{i=1}^{m} \left(2^{k_i} - 1\right). \qquad (1)$$

     where n is the population size, m is the number of linkage groups, k_i is the size of the i^{th} group.

   - **Compressed population complexity**, which quantifies the data compression in terms of the entropy of the marginal distribution over all partitions.

     $$C_p = n \sum_{i=1}^{m} \sum_{j=1}^{2^{k_i}} -p_{ij} \log_2\left(p_{ij}\right), \qquad (2)$$

     where p_{ij} is the frequency of the j^{th} gene sequence of the genes belonging to the i^{th} partition.

   Class-Search method: In eCGA, both the structure and the parameters of the model are searched and optimized to best fit the data. While the probabilities are learnt based on the variable instantiations in the population of selected individuals, a greedy-search heuristic is used to find an optimal or near-optimal probabilistic model. The search method starts by treating each decision variable as independent. The probabilistic model in this case is a vector of probabilities, representing the proportion of individuals among the selected individuals having a value `'1'` (or alternatively `'0'`) for each variable. The model-search method continues by merging two partitions that yields greatest improvement in the model-metric score. The subset merges are continued until no more improvement in the metric value is possible.

5. *Offspring creation:* New individuals are created by sampling the probabilistic model. The offspring population are generated by randomly generating subsets from the current individuals according to the probabilities of the subsets as calculated in the probabilistic model.

6. *Replacement:* We use two replacement techniques in this study: (1) Restricted tournament replacement (RTS) [23] in which offspring replaces the closest individual among w individuals randomly selected from the parent population, only if the offspring is better than the closest parent. (2) Elitist replacement used in NSGA-II, in which the parent and offspring population are combined. The domination ranks and crowding distances are computed on the combined population. Individuals with increasing ranks are gradually added starting from those with the lowest rank into the new population till its size reaches to n. However, if it is not possible to add all the solutions belonging to a particular rank without increasing the population size to greater than n, then individuals with greater crowding distances are preferred.

7. Repeat steps 2–6 until one or more termination criteria are met.

2.2 Test Problem

Our approach in verifying the performance of MOEDA is to consider bounding *adversarial problems* that exploit one or more dimensions of problem difficulty [24]. Particularly, we are interested in problems where building-block identification is critical for the GA success. Additionally, the problem solver (meCGA) should not have any knowledge of the building-block structure of the test problem, but should be known to researchers for verification purposes.

One such class of problems is the m-k deceptive *trap* problem, which consists of additively separable *deceptive* functions [25, 26, 27]. Deceptive functions are designed to thwart the very mechanism of selectorecombinative search by punishing any localized hillclimbing and requiring mixing of whole building blocks at or above the order of deception.

In this study, we use a class of test problems with two objectives: (1) m-k deceptive trap, and (2) m-k deceptive inverse trap. String positions are first divided into disjoint subsets or partitions of k bits each. The k-bit trap and inverse trap are defined as follows:

$$trap_k(u) = \begin{cases} 1 & \text{if } u = k \\ (1-d)\left[1 - \frac{u}{k-1}\right] & \text{otherwise} \end{cases} \quad (3)$$

$$invtrap_k(u) = \begin{cases} 1 & \text{if } u = 0 \\ (1-d)\left[\frac{u-1}{k-1}\right] & \text{otherwise} \end{cases}, \quad (4)$$

where u is the number of 1s in the input string of k bits, and d is the signal difference. Here, we use k = 3, 4, and 5, and d = 0.9, 0.75, and 0.8 respectively.

The m-k trap and inverse trap functions have conflicting objectives. Any solution that sets the bits in each partition either to 0s or 1s is Pareto optimal and thus there are a total of 2^m solutions in the Pareto-optimal front with $m + 1$ distinct points in the objective space.

2.3 Experimental methodology

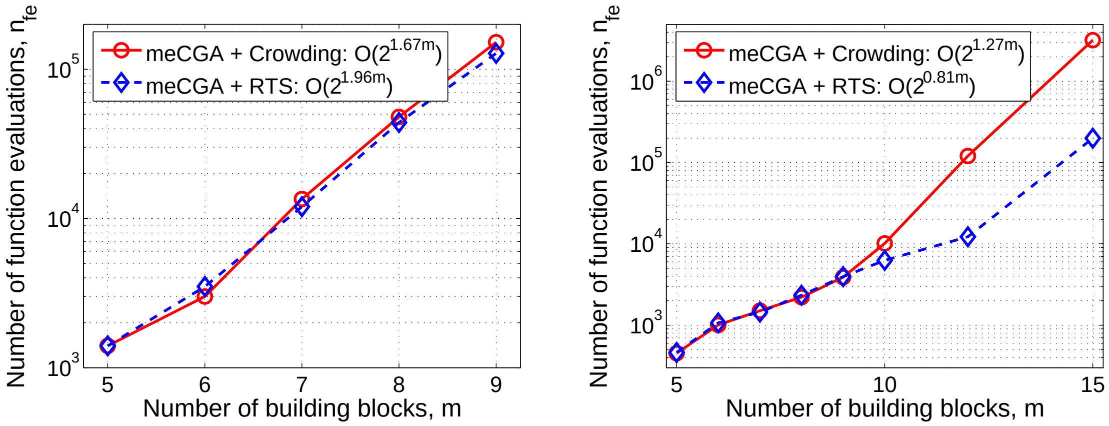
We measure the scalability of MOEDAs as the minimum number of function evaluations required to maintain at least one copy of all the Pareto-optimal solutions for problems of different sizes. For each problem type, problem size, and algorithm, a bisection method was used to determine a minimum population size to allocate at least one individual to each representative solution in the Pareto front. As mentioned earlier, for the test problems we consider in this study, for an ℓ-bit problem—where $\ell = m \cdot k$—there are 2^m Pareto-optimal solutions with $m + 1$ distinct objective value pairs. In this study, we investigate the population size required to (1) find at least one copy of all the 2^m Pareto-optimal solution, and (2) find at least one copy of the $m + 1$ distinct points in the Pareto-optimal front. That is, we consider Pareto-optimal solutions with the same values of both objectives to be equivalent.

The probability of maintaining at least one copy of all the representative Pareto-optimal solutions at a given population size is computed by averaging 10–30 independent MOEA runs. The minimum population size required to maintain at least one copy of all the representative solutions in the Pareto front are averaged over 10-30 independent bisection runs. Therefore, the results for each problem type, problem size, and algorithm correspond to 100–900 independent GA runs. The number of generations for meCGA was bounded by 5ℓ, where ℓ is the string length.

2.4 Scalability of meCGA

We have tried m-k deceptive trap and inverse trap functions for k = 3, 4, and 5, for brevity, we only show results for $k = 3$ in this paper. Moreover, we note that the results for other values of k are qualitatively similar and those for $k = 3$ are representative of the behavior of meCGA.

Figure 1(a), shows the scalability of meCGA with the problem size for m-3 deceptive trap and inverse trap problem. We plot the minimum number of function evaluations required to allocate at least one copy of all the solutions in the Pareto-optimal front. As shown in the figure,



(a) (b)

Figure 1: Scalability of meCGA with Crowding and with RTS for the m-3 deceptive trap and inverse trap with the problem size. Here, we plot the minimum number of function evaluations required to search and maintain at least one copy of (a) all the 2^m solutions in the Pareto-optimal front, and (b) only the $m+1$ solutions in the Pareto-optimal front with different objective-value pairs. Here, we treat the genotypically (and phenotypically) different Pareto-optimal solution with same values in both objectives to be equivalent.

all algorithms scale-up exponentially. The scale-up does not improve even if we restricted the requirement to finding only those $m+1$ Pareto-optimal solutions with different objective-value pairs as shown in Figure 1(b). That is, even if we consider genotypically (and phenotypically) distinct solutions that have the same value in both objectives to be equivalent, meCGA scales exponentially. This is despite the linkage information being identified correctly by meCGA. Additionally, the scalability does not improve if the niching or speciation is performed in the objective space (as in NSGA-II) or in the variable space (as in restricted tournament selection). To reiterate, meCGA scales up exponentially on the trap and inverse trap functions, in spite of accurate identification of the building blocks. Furthermore, when considering the m-k trap functions or inverse trap functions eCGA scales polynomially.

3 Exponential growth of Pareto-Optimal Solutions

As mentioned in the previous section, the exponential scale-up is not due to incorrect linkage identification and mixing [28, 29, 30], but because the niching mechanism gets quickly overwhelmed due to the exponential growth in the number of Pareto-optimal solutions. Furthermore, the distribution of the 2^m solutions in the Pareto-optimal front is not uniform. There are exponentially as many solutions in the middle of the front than at the edges (see table 1). That is, there is only one solution—a binary string with all 0s and all 1s—at each extreme of the Pareto-optimal front. In contrast, there are $\binom{m}{m/2} \approx \mathcal{O}(e^m)$ genotypically different solutions in the middle of the Pareto-optimal front with same values in both objectives.

$n_{1,BBs}$	0	1	$\cdots$	i	$\cdots$	m
$n_{0,BBs}$	m	$m-1$	$\cdots$	$m-i$	$\cdots$	0
# solutions	1	m	$\cdots$	$\binom{m}{i}$	$\cdots$	1

Table 1: Distribution of genotypically and phenotypically different solutions in the Pareto-optimal front with same values in both objectives. $n_{1,BBs}$ refers to the number of k-bit partitions (substructures) with 1s and $n_{0,BBs}$ is the number of k-bit partitions with 0s.

This highly non-linear distribution of solutions in the Pareto-front has two effects on the niching mechanisms used in MOEAs in general, and MOEDAs in particular:

- Since the extremes of the Pareto-optimal front (maximizing most partitions or sub-structures with respect to one particular objective) has exponentially smaller representatives than in the middle, it takes exponentially longer time, or exponentially larger population size [24, 30] to search and maintain the solutions at the extremes of the Pareto-optimal front. When the population size is fixed, the probability of maintaining a solution in the middle of the Pareto-optimal front is higher than doing so in extremes of the front, as shown in figure 2.
- Since there are multiple points that are genotypically and phenotypically different, but lie on the same point on the Pareto-optimal front (solutions have same values in both objectives), some of them vanish over time due to drift. The drift affects both the solutions in the middle and the extremes of Pareto front.

3.1 Overwhelming the Niching Method

To illustrate, how additively decomposable problems with conflicting objectives can overwhelm the niching mechanism used in MOEAs—irrespective of linkage adaptation

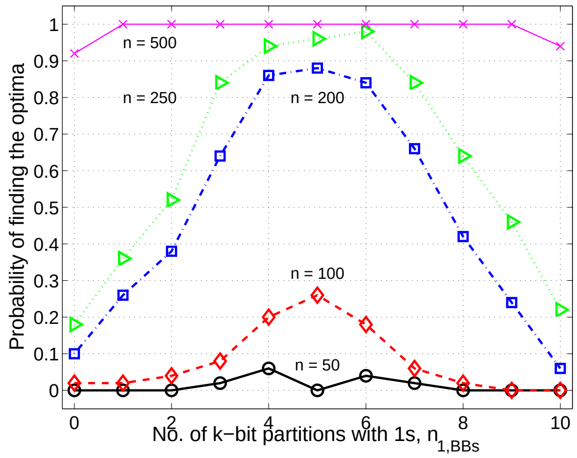


Figure 2: Probability of finding and maintaining different solutions on the Pareto-optimal for the 10-3 deceptive trap and inverse trap problem as a function of population size. The results are for meCGA with elitist crowding and the results are averaged over 100 independent runs.

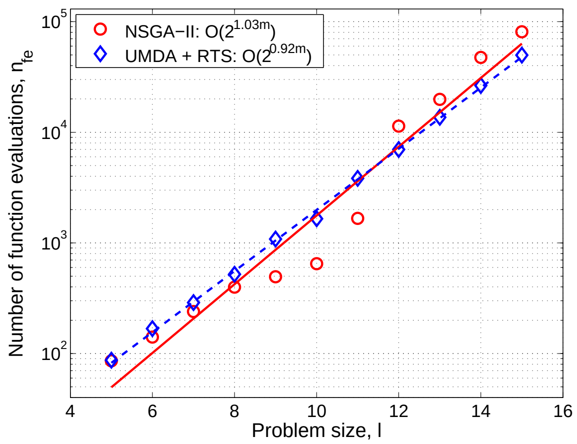


Figure 3: Scalability of NSGA-II and mUMDA on the OneMax-ZeroMax problem in terms of minimum number of function evaluations required to maintain at least on copy of each of the $\ell + 1$ distinct solutions in the Pareto-optimal front. Both algorithms with two different niching methods scale-up exponentially with the problem size.

capabilities of the evolutionary algorithm—and lead to exponential scalability, we consider a problem where linkage learning is not required. Specifically, we consider the OneMax-ZeroMax problem which is similar to bi-criteria OneMax problem of Chen [4]. In OneMax-ZeroMax problem, the task is to maximize two objectives, one which is the sum of all the bits with value 1, and the other is the sum of all the bits with value 0:

$$f_{\text{OneMax}}(X) = \sum_{i=1}^{\ell} x_i, \tag{5}$$

$$f_{\text{ZeroMax}}(X) = \sum_{i=1}^{\ell} (1 - x_i), \tag{6}$$

where ℓ is the problem size, and x_i is the value of the i^{th} bit of a candidate solution X.

We specifically choose the OneMax-ZeroMax problem to isolate the effect of linkage identification or lack there of from those of the niching methods on the scalability of the MOEAs. Unlike the m-k deceptive trap and inverse trap function, linkage identification is not necessary for the OneMax-ZeroMax problem. Furthermore, both OneMax and ZeroMax problems are GA-easy problems which a simple selectorecombinative GA with uniform crossover and tournament selection can solve in linear time [31, 32].

However, in a multiobjective scenario of the OneMax-ZeroMax, the *entire* search space (2^ℓ) belongs to the Pareto-optimal front with $\ell + 1$ distinct objective-value pairs. Therefore, in order to maintain all the Pareto-optimal solutions, we would require $\mathcal{O}(2^\ell)$ population size. From the details presented in the previous sections, even if we relax the scalability requirement to finding at least one copy of all $\ell + 1$ distinct Pareto-optimal solutions, the exponential requirement in the population size (and consequently the number of function evaluations) is not relaxed. Therefore, as expected, the MOEAs, particularly multiobjective univariate marginal distribution algorithm (mUMDA) and NSGA-II, scale-up exponentially in solving the OneMax-ZeroMax problem as shown in figure 3. The mUMDA algorithm used in this study is identical to meCGA where the probabilistic model is a univariate model where each variable/bit is considered independent to each other [33], which is the ideal model for the OneMax-ZeroMax problems.

The results clearly indicate how the niching methods—both those that work in parameter space (RTS) and those that work in objective space (Crowding)—get overwhelmed due to exponentially large number of solutions in the Pareto-optimal front. That is, since the number of Pareto-optimal solutions grow exponentially, in order to maintain at least one copy of all the global solutions we would require exponentially large population sizes. Additionally, the results also show that even if the requirement is relaxed by treating all the different points that lie on the same point in the Pareto-optimal front to be equivalent, the scale-up does not improve. Finally, the results suggest that in decomposable problems, if all or majority of the sub-structures compete in the two objectives, then the niching method fails to maintain good coverage, leading to exponential scale-up.

This combinatorial growth in the number of Pareto-optimal solutions is a deal breaker for tractable solutions

- Acknowledge that with practical population sizes, some of the Pareto-optimal solutions cannot be covered (especially at the edges of the Pareto front), and do the best we can. In such a scenario, an MOEA with linkage-adaptation capabilities outperforms MOEAs with fixed recombination operators [34].
- Size the population appropriately in accordance with the exponential growth in the Pareto-optimal solutions. Here, Mahfoud's population-sizing model for niching methods [35], which predicts that the population size grows linearly with the number of global solutions, is applicable.

- Understand the fundamental limits on the growth in the number of Pareto-optimal solutions and thereby the type of search problems that permit tractable search. If we want MOEDAs to scale polynomially on additively-decomposable problems, the number of Pareto-optimal solutions have to be limited. In other words, if the number of sub-structures that are different (or compete) between multiple objectives is limited, then MOEDAs can scale polynomially. That is, There is an imposed limit on the type of additively decomposable problems MOEDAs can solve in polynomial time. We use facetwise models to predict this limit on the growth of competing sub-structures between multiple objectives (and consequently, the number of Pareto-optimal solutions) in the next section.

4 Limit on the Growth of Competing Sub-Structures

The results in the previous two sections clearly indicate that MOEDAs with either RTS or crowding mechanism of NSGA-II scale-up exponentially with problem size on boundedly-difficult additively-separable multiobjective problems. We also demonstrated that the exponential scalability is due to the niching method being overwhelmed because of exponentially large number of solutions in the Pareto-optimal front. The exponential growth in the number of Pareto-optimal solutions imposes a fundamental limitation on the type of problems which MOEDAs can solve in polynomial time. That is, MOEDAs can solve only those multiobjective problems in polynomial time, which have limited growth in the number of Pareto-optimal solutions.

One way to restrict the growth of the Pareto-optimal solutions is to control the number of sub-structure (building blocks) that compete between the two objectives, m_d. That is, for a problem with m sub-structures, the two objectives differ in only m_d sub-structures and share the same $m - m_d$ sub-structures. For example, consider $4 - bit$ OneMax-ZeroMax problem, $m_d = 2$, and without loss of generality, that the last two building blocks differ between the two objectives. Then there are only four Pareto-optimal solutions (as opposed to 2^4 = 16): `1100`, `1101`, `1110`, and `1111`. Since the total number of Pareto-optimal solutions, $n_{opt} = 2^{m_d}$, by controlling the number of competing sub-structures, we implicitly control the total number of Pareto-optimal solutions.

The growth-rate of the competing sub-structures should be such that the effect of model accuracy, decision making, and building-block supply on the population sizing is dominant over the effect of niching on the population size. The effect of model accuracy, decision making and sub-structure supply on the population sizing of eCGA is given by [20, 19]:

$$n_{\text{eCGA}} \propto c_1 \cdot 2^k \cdot m \log m, \tag{7}$$

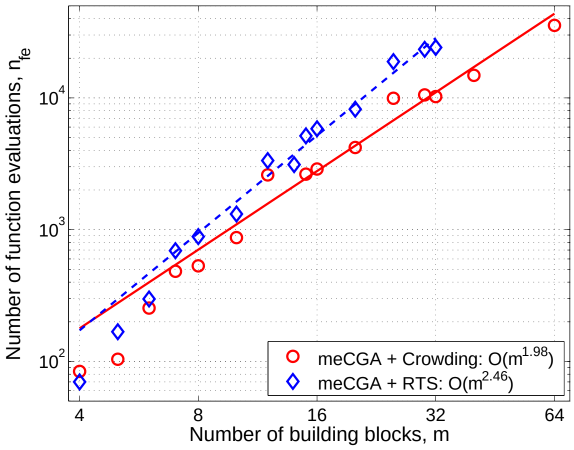


Figure 4: The scalability of meCGA with the crowding mechanism of NSGA-II and RTS niching for both OneMax-ZeroMax and m-3 deceptive trap and inverse trap problems. The growth rate of number of sub-structures that compete in the two objectives for a given problem size is controlled as given by equation 10.

where, c_1 is a constant. The effect of niching method on the population-sizing of GAs was modeled by Mahfoud [35] and is reproduced below:

$$n_{\text{niching}} \propto \frac{\log\left[\left(1-\gamma^{1/t}\right)/n_{opt}\right]}{\log\left[\left(n_{opt}-1\right)/n_{opt}\right]} \approx c_2 \cdot 2^{m_d}, \tag{8}$$

where γ is the probability of maintaining at least one copy of all the Pareto-optimal solutions, t is the number of generations we need to maintain all the niches, and c_2 is a constant. While Mahfoud derived the population-sizing estimate for fitness-sharing method, it is generally applicable to other niching methods and MOEAs as well [36, 13].

In order to restrict the number of Pareto-optimal solutions, and thereby to circumvent the niching method from being overwhelmed we require $n_{\text{eCGA}} \geq n_{\text{niching}}$. That is,

$$c_2 \cdot 2^{m_d} \geq c_1 \cdot 2^k \cdot m \log m. \tag{9}$$

The above equation can be approximated [1] to obtain a conservative estimate of the maximum number of competing sub-structures that circumvent the niching mechanism from being overwhelmed is given by:

$$m_d \approx k + \log_2(m) \tag{10}$$

To reiterate, MOEDAs can solve additively separable problems in polynomial time if the number of building blocks that differ between two objectives is less than that predicted in the above equation. We verify this assertion with empirical results for the OneMax-ZeroMax problem in Figure 4. Specifically, for the $m - m_d$ building blocks shared by both objectives we consider the OneMax function, and for the m_d differing building blocks, we consider the OneMax-ZeroMax problem. The particular building blocks that differ between the two objectives are randomly

---

[1] Since $\log_2 \frac{c_1 \log m}{c_2} \sim 1$, we neglect the term

chosen for a particular problem instance. As shown in figure 4, the results indicate that when the limit on the growth-rate of competing sub-structures is satisfied, the MOEDAs scale-up polynomially with the problem size.

5 Summary and Conclusions

In this paper, we studied the scalability of multiobjective estimation of distribution algorithms (MOEDAs), specifically multiobjective extended compact genetic algorithm (meCGA), on a class of boundedly-difficult additively separable problems. We observed that even when the linkages were correctly identified, the MOEDAs scaled-up exponentially with problem size due to the combinatorial growth in the number of Pareto-optimal solutions. We demonstrated that even if the linkage is correctly identified, massive multimodality of the search problems can easily overwhelm the nicher and lead to exponential scale-up. That is, in decomposable problems, if majority or all the sub-structures compete in different objectives, then the number of Pareto-optimal solutions increase exponentially. This exponential increase overwhelms the nicher and causes significant problems in maintaining a good coverage of the Pareto-optimal front. This combinatorial explosion of Pareto-optimal solutions introduces a fundamental limit on the number of competing sub-structures between multiple objectives. Using facetwise models that incorporate the combined effects of model accuracy, decision making, and sub-structure supply, and the effect of niching on the population sizing, we predict this limit on the growth rate of maximum number of sub-structures that can compete in the two objectives to circumvent the failure of the niching method. If the number of competing sub-structures between the multiple objectives is less than the proposed limit, MOEDAs scale-up polynomially with the problem size on boundedly-difficult problems.

Acknowledgments

This work was sponsored by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant F49620-03-1-0129, the National Science Foundation under ITR grant DMR-03-25939 at Materials Computation Center, and ITR grant DMR-01-21695 at CPSD, and the Dept. of Energy under grant DEFG02-91ER45439 at Fredrick Seitz Materials Research Lab. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright notation thereon.

Bibliography

[1] K. Deb. *Multi-objective optimization using evolutionary algorithms*. John Wiley and Sons, Chichester, UK, 2001.

[2] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont. *Evolutionary algorithms for solving multi-objective problems*. Kluwer Academic Publishers, Boston, MA, 2002.

[3] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7(3):205–230, 1999.

[4] J.-H. Chen. *Theory and Applications of Efficient Multi-objective Evolutionary Algorithms*. PhD thesis, Feng Chia University, Taichung, Taiwan, 2004.

[5] M. Pelikan, F. Lobo, and D. E. Goldberg. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21:5–20, 2002. (Also IlliGAL Report No. 99018).

[6] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.

[7] D. E. Goldberg. The race, the hurdle, and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity. In P. Bentley, editor, *Evolutionary Design by Computers*, chapter 4, pages 105–118. Morgan Kaufmann, San Mateo, CA, 1999.

[8] P. A. N. Bosman and D. Thierens. Multiobjective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *International Journal of Approximate Reasoning*, 31(3):259–289, 2002. (Also Tech. Report No. UU-CS-2001-59).

[9] N. Khan, D. E. Goldberg, and M. Pelikan. Multiobjective Bayesian optimization algorithms. *Proceedings of the 2002 Genetic and Evolutionary Computation Conference*, page 684, 2002. (Also IlliGAL Report No. 2002009).

[10] J. Ocenasek. *Parallel estimation of distribution algorithms*. PhD thesis, Brno University of Technology, Brno, Czech Republic, 2002.

[11] C. W. Ahn. *Theory, Design, and Application of Efficient Genetic and Evolutionary Algorithms*. PhD thesis, Gwangju Institute of Science and Technology, Korea, 2005.

[12] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. (Also KanGAL Report No. 2000001).

[13] N. Khan. Bayesian optimization algorithms for multiobjective and hierarchically difficult problems. Master's thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002. (Also IlliGAL Report No. 2003021).

[14] P. A. N. Bosman and D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188, 2002.

[15] M. Laumanns and J. Ocenasek. Bayesian optimization algorithm for multi-objective optimization. *Parallel Problem Solving from Nature*, pages 298–307, 2002.

[16] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Linkage learning, estimation distribution, and Bayesian networks. *Evolutionary Computation*, 8(3):314–341, 2000. (Also IlliGAL Report No. 98013).

[17] P. A. N. Bosman and D. Thierens. Linkage information processing in distribution estimation algorithms. *Proceedings of the 1999 Genetic and Evolutionary Computation Conference*, pages 60–67, 1999. (Also Tech. Report. No. UU-CS-1999-10).

[18] P. A. N. Bosman and D. Thierens. Mixed ideas. Technical Report UU-CS-2000-45, Utrecht University, 2000.

[19] 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.

[20] M. Pelikan, K. Sastry, and D. E. Goldberg. Scalability of the Bayesian optimization algorithm. *International Journal of Approximate Reasoning*, 31(3):221–258, 2003. (Also IlliGAL Report No. 2002024).

[21] G. Harik. Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Urbana, IL, January 1999.

[22] J. J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

[23] G. R. Harik. Finding multimodal solutions using restricted tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31, 1995. (Also IlliGAL Report No. 94002).

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

[25] D. H. Ackley. *A Connectionist Machine for Genetic Hill Climbing*. Kluwer Academic Publishers, 1987.

[26] 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.

[27] K. Deb and D. E. Goldberg. Analyzing deception in trap functions. *Foundations of Genetic Algorithms*, 2:93–108, 1992. (Also IlliGAL Report No. 91009).

[28] D. E. Goldberg, D. Thierens, and K. Deb. Toward a better understanding of mixing in genetic algorithms. Journal of the Society of Instrument and Control Engineers, 32(1):10–16, 1993. (Also IlliGAL Report No. 92009).

[29] D. Thierens and D. E. Goldberg. Mixing in genetic algorithms. Proceedings of the Fifth International Conference On Genetic Algorithms, pages 38–45., 1993.

[30] D Thierens. Scalability problems of simple genetic algorithms. Evolutionary Computation, 7(4):331–352., 1999.

[31] H. Mühlenbein and D. Schlierkamp-Voosen. Predictive models for the breeder genetic algorithm: I. continous parameter optimization. *Evolutionary Computation*, 1(1):25–49, 1993.

[32] 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).

[33] Heinz Mühlenbein and Gerhard Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. *Parallel Problem Solving from Nature*, 4, 1996.

[34] M. Pelikan, K. Sastry, and D. E. Goldberg. Multiobjective hBOA, clustering, and scalability. IlliGAL Report No. 2005005, University of Illinois at Urbana-Champaign, Urbana, IL, February 2005.

[35] S. W. Mahfoud. Population size and genetic drift in fitness sharing. *Foundations of Genetic Algorithms*, 3:185–224, 1994. (Also IlliGAL Report No. 94005).

[36] P. Reed. *Striking the balance: Long-term groundwater monitoring design for multiple conflicting objectives*. PhD thesis, University of Illinois, Urbana, IL, 2002.