

Selection, Drift, Recombination, and Mutation in Multiobjective Evolutionary Algorithms on Scalable MNK-Landscapes

Hernán E. Aguirre and Kiyoshi Tanaka

Faculty of Engineering, Shinshu University
4-17-1 Wakasato, Nagano, 380-8553 JAPAN
{ahernan, ktanaka}@gipwc.shinshu-u.ac.jp

Abstract. This work focuses on the working principles, behavior, and performance of state of the art multiobjective evolutionary algorithms (MOEAs) on discrete search spaces by using MNK-Landscapes. Its motivation comes from the performance shown by NSGA-II and SPEA2 on epistatic problems, which suggest that simpler population-based multiobjective random one-bit climbers are by far superior. Adaptive evolution is a search process driven by selection, drift, mutation, and recombination over fitness landscapes. We group MOEAs features and organize our study around these four important and intertwined processes in order to understand better their effects and clarify the reasons to the poor performance shown by NSGA-II and SPEA2. This work also constitutes a valuable guide for the practitioner on how to set up its algorithm and gives useful insights on how to design more robust and efficient MOEAs.

1 Introduction

Epistasis in the context of evolutionary algorithms (EAs) describes nonlinearities in fitness functions due to changes in the values of interacting bits. Epistasis is recognized as an important factor that makes a problem difficult for optimization algorithms and its influence on the performance of single objective EAs is being increasingly investigated. Particularly, Kauffman’s NK-Landscapes model of epistatic interactions [1] has been the center of several studies, both for the statistical properties of the generated landscapes and for their *EA-hardness*. See for example [2–5] and there in. Studies on the behavior of single objective EAs on NK-Landscapes have proved useful to advance our understanding of EA’s working principles and served to design robust and better algorithms [5].

Contrary to single objective EAs, studies concerning epistasis within the context of multiobjective evolutionary algorithms (MOEAs) are few and its effects still not well understood. Recently, Aguirre and Tanaka [6] have extended Kauffman’s NK-Landscapes model of epistatic interactions to multiobjective MNK-Landscapes, giving insights into their properties in order to understand how the parameters of the landscapes relate to multiobjective concepts such as shape of the fronts, number of non-dominated fronts, number of non-dominated solutions,

accessibility to the true Pareto front, correlation between and within fronts, and metrics. From a multiobjective random test problem generator standpoint [7], desirable features of MNK-Landscapes are that the problems are easy to construct and can scale to any number of objectives M , number of bits N , and number of epistatic interactions K , allowing the creation of sub-classes of combinatorial non-linear problems for discrete search spaces in which we can test the working principles of MOEAs in order to design better and more robust algorithms. Aguirre and Tanaka have also studied the behavior of multiobjective random one-bit climbers (moRBCs) [8] on MNK-Landscapes and have provided initial results on the performance of two well known representatives of the latest generation of elitist MOEAs [9], namely NSGA-II [10] and SPEA2 [11].

This work focuses on the working principles, behavior, and performance of state of the art MOEAs on discrete search spaces by using MNK-Landscapes. Its motivation comes from the performance shown by NSGA-II and SPEA2 on epistatic problems [9, 8], which suggest that simpler population-based moRBCs are by far superior. Adaptive evolution is a search process driven by selection, drift, mutation, and recombination over fitness landscapes [1]. We group MOEAs features and organize our study around these main processes. In most of the latest generation MOEAs [10, 12] selection incorporates elitism and it is biased by Pareto dominance and a diversity preserving strategy in objective space. Genetic operators vary according to whether the search space is continuous or discrete. In discrete search spaces, like MNK-Landscapes, recombination is usually implemented as one-point or two-point crossover and mutation as the standard bit flipping method. Some approaches also include specialized mutation operators to perform local search. In addition to these features explicit to the algorithm design, drift is also an important process that drives evolution and it is implicit to all stochastic algorithms working on finite small populations, although sometimes highly overlooked. In this paper we study the effects of these important and intertwined processes in order to understand them better, clarifying the reasons to the poor performance shown by NSGA-II and SPEA2. This work also constitutes a valuable guide for the practitioner on how to set up its algorithm and gives useful insights on how to design more robust and efficient MOEAs.

2 Multiobjective MNK-Landscapes

A multiobjective MNK-Landscape is defined as a vector function mapping binary strings into real numbers $\mathbf{f}(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_M(\cdot)) : \mathcal{B}^N \rightarrow \mathbb{R}^M$, where M is the number of objectives, $f_i(\cdot)$ is the i -th objective function, $\mathcal{B} = \{0, 1\}$, and N is the bit string length. $\mathbf{K} = \{K_1, \dots, K_M\}$ is a set of integers where K_i ($i = 1, 2, \dots, M$) is the number of bits in the string that epistatically interact with each bit in the i -th landscape. Each $f_i(\cdot)$ can be expressed as an average of N functions as follows

$$f_i(\mathbf{x}) = \frac{1}{N} \sum_{j=1}^N f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \dots, z_{K_i}^{(i,j)}) \quad (1)$$

where $f_{i,j} : \mathcal{B}^{K_i+1} \rightarrow \mathbb{R}$ gives the fitness contribution of bit x_j to $f_i(\cdot)$, and $z_1^{(i,j)}, z_2^{(i,j)}, \dots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with bit x_j in the string \mathbf{x} . The fitness contribution $f_{i,j}$ of bit x_j is a number between $[0.0, 1.0]$ drawn from a uniform distribution. Thus, each $f_i(\cdot)$ is a non-linear function of \mathbf{x} expressed by a Kauffman's NK-Landscape model of epistatic interactions [1].

For a given N , we can tune the ruggedness of the fitness function $f_i(\cdot)$ of the i -th objective by varying K_i . In the limits, $K_i = 0$ corresponds to a model in which there are no epistatic interactions and the fitness contribution from each bit value is simply additive, which yields a single peaked smooth i -th fitness landscape. On the opposite extreme, $K_i = N - 1$ corresponds to a model in which each bit value is epistatically affected by all the remaining bit values yielding a maximally rugged fully random i -th fitness landscape. Varying K_i from 0 to $N - 1$ gives a family of increasingly rugged multi-peaked landscapes.

Besides defining N and K_i for each $f_i(\cdot)$, it is also possible to arrange the epistatic pattern between bit x_j and the K_i other interacting bits. That is, the distribution $D_i = \{\text{random}, \text{nearest neighbor}\}$ of K_i bits among N . Thus, M , N , $\mathbf{K} = \{K_1, K_2, \dots, K_M\}$, and $\mathbf{D} = \{D_1, D_2, \dots, D_M\}$, completely specify a multiobjective MNK-Landscape. By varying these parameters we can analyze the properties of the multiobjective landscapes and study the effects of the number of objectives, size of the search space, intensity of epistatic interactions, and epistatic pattern on the performance of multiobjective optimization algorithms on combinatorial discrete search spaces.

3 The Algorithms

In this work we present results by NSGA-II [10], SPEA2 [11], and moRBC($\delta : 1+1$) [8], a multiobjective random one-bit climber using a population for restarts. Since we refer extensively to NSGA-II and moRBC($\delta : 1+1$) is an important reference for comparison we include a brief description of both algorithms.

3.1 NSGA-II

NSGA-II keeps at the t -th generation a parent population \mathcal{P}_t and an offspring population \mathcal{Q}_t , both of same size μ . The parent population \mathcal{P}_{t+1} at the $t+1$ -th generation is a subset of the best individuals obtained by truncating the combined population of parents and offspring $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{Q}_t$. That is, $\mathcal{P}_{t+1} \subset \mathcal{R}_t$, where $|\mathcal{R}_t| = 2\mu$ and $|\mathcal{P}_{t+1}| = \mu$. To obtain \mathcal{P}_{t+1} , \mathcal{R}_t is first classified into non-dominated fronts. The first front \mathcal{F}_1 contains the best non-dominated solutions \mathcal{S}_1 . The subsequent fronts $\mathcal{F}_j, j > 1$, contain lower level non-dominated solutions and are obtained by disregarding solutions corresponding to the previous higher non-dominated fronts, i.e. $\mathcal{F}_j, j > 1$, is obtained from the set $\mathcal{R}_t - \bigcup_{k=1}^{j-1} \mathcal{S}_k$. Once the classification of non-dominated fronts is over, the parent population \mathcal{P}_{t+1} is filled with solutions belonging to the higher fronts, starting with front \mathcal{F}_1 . If the whole front \mathcal{F}_i does not fit, the required number of individuals with best crowding distance are selected to fill the parent population. Each solution in \mathcal{P}_t

is assigned a rank (fitness) equal to its nondomination level (1 is the best level). Binary tournament selection with crowded tournament operator, recombination, and mutation operators are used to create the offspring population Q_{t+1} from P_{t+1} . During selection, solution \mathbf{x} wins a tournament if it has a better rank than \mathbf{y} . If \mathbf{x} and \mathbf{y} have the same rank, the solution with best crowding distance wins.

3.2 moRBC($\delta : 1 + 1$)

moRBC($\delta : 1 + 1$) is a random one-bit climber that at all times keeps one parent individual from which it creates one offspring. It begins with a randomly created parent string of length N . Then, a random permutation π of the string positions is generated. A child is created by cloning the parent and flipping the bit at position π_i , the child is evaluated and replaces the parent if it *dominates* the parent. Child creation, evaluation, and (possibly) parent replacement are repeated for all π_i , $1 \leq i \leq N$. If no parent replacements were detected a *dominance local optimum* has been found and moRBC($\delta : 1 + 1$) RESTARTS the search. Testing continues by going back to create a new permutation π . This process ends once a given number of evaluations has been expended. A *Population* of up to δ solutions non-dominated by the parent and amongst themselves are kept during the process. moRBC($\delta : 1 + 1$) RESTARTS the search by replacing the parent with one individual chosen from the collected *Population*. If *Population* is empty, the parent is replaced with a random string created anew. Additionally, the non-dominated solutions found throughout the search are kept in an *Archive* of limited capacity. The procedures that update the *Population* and the *Archive* use NSGA-II's diversity preserving mechanism in objective space, where non-dominated individuals with better crowding distance [10](p.236) are preferred in case the *Population/Archive* has reached its capacity. Duplicate solutions are not allowed in the *Population* or in the *Archive*.

4 Metric, Test Problems, and Parameters

In this work we use the hypervolume metric \mathcal{H} proposed by Zitzler [13] to evaluate and compare the performance of the algorithms. Let \mathcal{A} be a set of non-dominated solutions. The metric \mathcal{H} calculates the volume of the M-dimensional region in objective space enclosed by the elements of \mathcal{A} and a dominated reference point, hence computing the size of the region \mathcal{A} dominates. The hypervolume can be expressed as

$$\mathcal{H}(\mathcal{A}) = \bigcup_{i=1}^{|\mathcal{A}|} (\mathcal{V}_i - \bigcap_{j=1}^{i-1} \mathcal{V}_i \mathcal{V}_j) \quad (2)$$

where \mathcal{V}_i is the hypervolume rendered by the point $\mathbf{x}_i \in \mathcal{A}$ and the reference point. The hypervolume is among the few recommended metrics for comparing non-dominated sets [14] and there is some theoretical evidence [15] that the maximization of the hypervolume constitutes the necessary and sufficient condition for the solutions in objective space to be maximally diverse Pareto optimal solutions of a discrete, multiobjective, optimization problem. The reference point to calculate the hypervolume is set to $[0.0, \dots, 0.0]$.

In our study we use MNK-Landscapes with $M = \{2, 3, 5\}$ objectives, $N = \{20, 50, 100\}$ bits, vary the number of epistatic interactions from %0 to %50 of N simultaneously in all objectives ($K_1, \dots, K_M = K$), and set *random* epistatic patterns among bits for all objectives ($D_1, \dots, D_M = \text{random}$). For each combination of M , N and K , 50 different problems randomly generated are employed.

NSGA-II and SPEA2 use a population size of 100 individuals, two point crossover for recombination with probability $p_c = 0.6$, and bit flipping mutation with probability $p_m = 1/N$ per bit. moRBC($\delta : 1 + 1$) also uses a population $\delta = 100$ individuals. For all algorithms, the number of evaluations is set to 3×10^5 and the *Archive* size is set to 100.

5 Performance by conventional NSGA-II and SPEA2

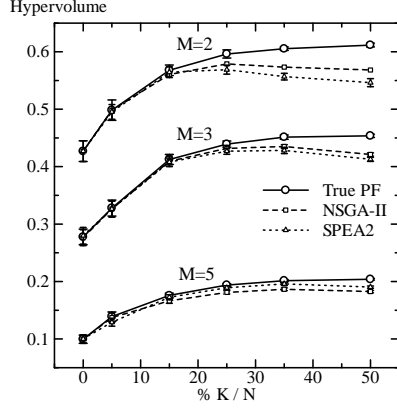
First, we present results by conventional NSGA-II and SPEA2 on scalable MNK-Landscapes for various values of M , N , and K , in order to have a broad view of the performance of these algorithms on combinatorial multiobjective epistatic problems. **Fig. 1** plots the Archive’s average hypervolume over the number of epistatic interactions K for $N = \{20, 50, 100\}$ bits landscapes. The average hypervolume of the true Pareto front obtained by enumeration is also included for $N = 20$ bits landscapes. Vertical bars overlaying the mean hypervolume curves represent 95% confidence intervals.

From **Fig. 1 (a)**, note that on $N = 20$ bits landscapes the trend of the hypervolume of the true Pareto front for any value of M is to rapidly increase with K , from $K = 0$ to small values of K , and to remain high for medium and large K . A similar trend is expected for the hypervolume of the true Pareto front on landscapes with higher values of N . Looking at results by NSGA-II and SPEA2, in **Fig. 1 (a)** we can see that the hypervolume of the solutions found by these algorithms approach the hypervolume of true Pareto front on $N = 20$ bits landscapes only for $K \leq 15\%N$. Increasing the number of bits N , we see that the value of the hypervolume of the non-dominated solutions found by the algorithms decreases continuously from $K \geq 8\%N$ for $N = 50$ and from $K \geq 5\%N$ for $N = 100$ bits. See **Fig. 1 (b)** and **Fig. 1 (c)**, respectively. These decreasing values are against the expected trend of the hypervolume of the true Pareto front and indicate that the search performance of the algorithms is worsening significantly as K increases.

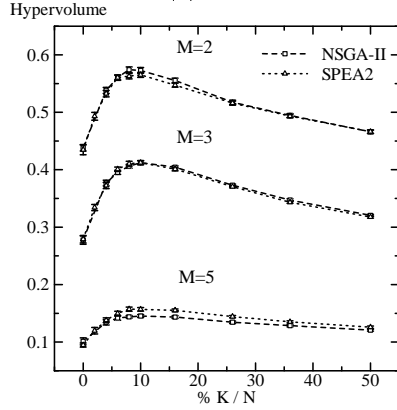
In the following we focus on NSGA-II and especially look into the effects of selection, drift, recombination, and mutation.

6 Selection and Drift

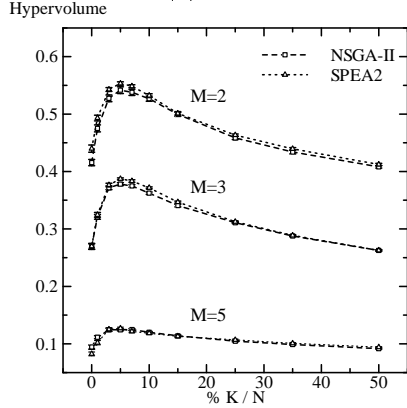
The main processes that drive evolution are selection, drift, mutation, and recombination. In this section we observe the effects of selection and drift, which decrease genetic variation being the homogenization of the population an extreme consequence of it. Selection features are made explicit during the design of the algorithms. In most of the latest generation MOEAs selection incorporates



(a) $N = 20$ bits

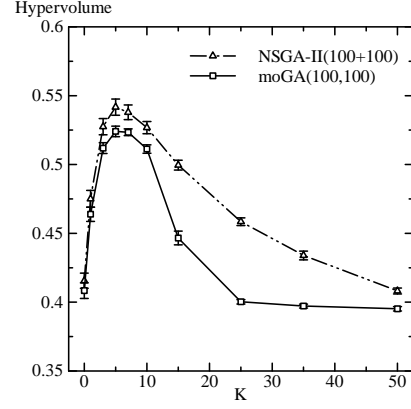


(b) $N = 50$ bits

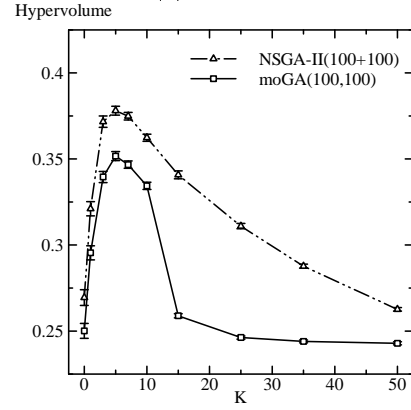


(c) $N = 100$ bits

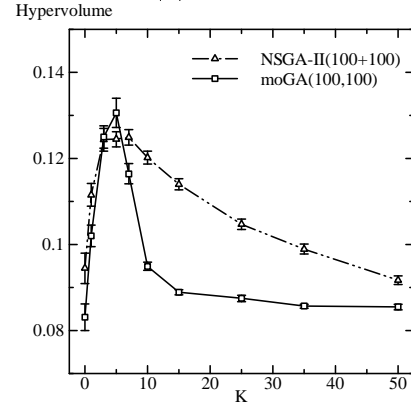
Fig. 1. Average hypervolume $\bar{\mathcal{H}}$ over number of epistatic interactions K by conventional NSGA-II and SPEA2.



(a) $M = 2$



(b) $M = 3$



(c) $M = 5$

Fig. 2. Effect of elitism. $\bar{\mathcal{H}}$ over K by conventional NSGA-II($\mu + \lambda$) using elitism and moGA(μ, λ) without elitism. $N=100$ bits landscapes.

elitism and it is biased by Pareto dominance and a diversity preserving strategy in objective space. On the other hand, drift is a process implicit to all stochastic algorithms working on finite small populations.

6.1 Elitism

Elitism is considered an important component of the selection process in state of the art evolutionary multiobjective optimizers. In order to have a clear idea of its contribution to the performance of MOEAs on epistatic problems this section compare results by NSGA-II($\mu+\lambda$) and moGA(μ,λ). NSGA-II implements elitism by keeping for the next generation the best (μ) individuals from the joined population ($\mu+\lambda$) of parents and offspring. On the other hand, moGA(μ,λ) replaces the parent population (μ) by its offspring population (λ) at each generation. moGA(μ,λ)'s other features are the same used by NSGA-II.

Fig. 2 shows results by NSGA-II(100+100) and moGA(100,100) for $M = \{2, 3, 5\}$ objectives on $N = 100$ bits landscapes. From this figure we can see that if elitism is not included there is a severe deterioration in performance for all values of K and M , except for $M = 5$ and $K = 5$. Note that the performance by moGA(100,100) falls sharply from $K = 10\%N$ to $K = 15\%N$ in $M = 2$ objectives. The fall in performance is even more pronounced for $M = 3$ and $M = 5$ objectives. These results are in accordance with the expectation that elitism is a very important feature for multiobjective combinatorial optimization. However, elitism can also bring about undesired side effects that could severely affect the efficacy and efficiency of the algorithms. Throughout the following sections we discuss some of them.

6.2 Genetic Drift

Genetic drift is a phenomenon that emerges from the stochastic operators of selection, recombination, and selection. It refers to the change on bit (allele) frequencies due to chance alone especially in small populations. In single objective EAs it is well known that genetic drift is one important factor that affects negatively the performance of EAs especially if a strong selection pressure is used, such as truncated selection (μ,λ) without elitism where $\mu < \lambda$. See [5], for example. The presence of elitism, for instance in the form of truncated selection ($\mu + \lambda$) used by NSGA-II, would increase selection pressure making elitist algorithms even more prone to the effects of drift.

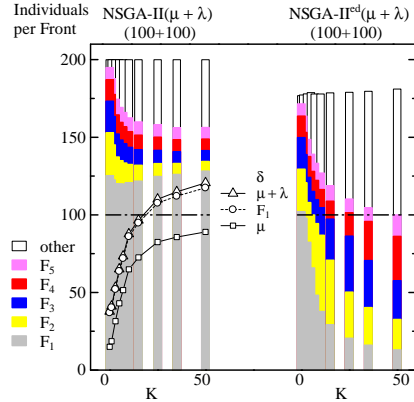
In this section we enhance NSGA-II by preventing fitness duplicates from the population in order to observe the effect of genetic drift on the performance of the algorithm. In the enhanced algorithm, called NSGA-II^{ed}, if several individuals have exactly the same fitness in all objectives then one is chosen at random and kept. The other equal fitness individuals are eliminated from the population. Fitness duplicates elimination is carried out before truncating the population from ($\mu + \lambda$) to (μ) individuals. This process aims to effectively eliminate clones without the need to compare hamming distances, postpone genetic drift, and remove an unwanted source of selective bias.

To explain and understand better the effects of duplicates on the performance of multiobjective algorithms, **Fig. 3** shows cumulative bar diagrams of the average number of individuals per non-dominated front over the number of epistatic interactions K . Results by the conventional NSGA-II and by the enhanced NSGA-II^{ed} that eliminates duplicates are presented for the top five fronts in $M = \{2, 3, 5\}$ objectives. In the case of conventional NSGA-II, the figure also shows with lines the number of duplicate individuals δ in the whole population $(\mu + \lambda)$ before truncation, in the first non-dominated front (F_1) also before truncation, and in the truncated population (μ). For NSGA-II^{ed} results are presented after elimination of duplicates. Horizontal lines indicate the truncation site.

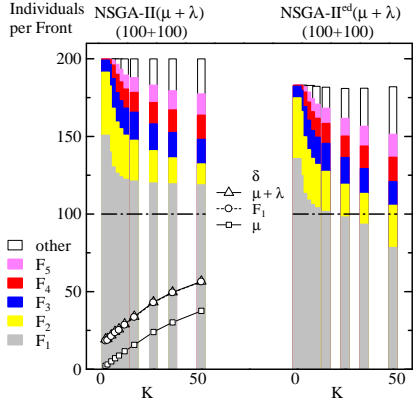
Looking at results by NSGA-II in **Fig. 3**, the following observations are relevant. (i) The number of duplicates increases as we increase the epistatic interactions K . (ii) The presence of duplicates reduces increasing the number of objectives M . (iii) Most duplicates belong to the first non-dominated front and a large number remain after truncation, especially for large K . Conversely, looking at the size of the cumulative bars by NSGA-II^{ed} in **Fig. 3**, we can deduce that the average number of duplicates eliminated at each generation by NSGA-II^{ed} is only a small fraction of the whole $(\mu + \lambda)$ population and it is similar for all K and M . For example, for $M = 2$ note that in NSGA-II the number of duplicates augment from 8% to 90% of the truncated population (μ) and from 18% to 60% of the whole population $(\mu + \lambda)$ increasing K from 0 to 50, respectively. By contrast, the average number of duplicates in NSGA-II^{ed} is around 9% for all K . The number of duplicates observed in NSGA-II^{ed} could be taken as the homogenization effect of drift and selection at each generation, whereas the number of duplicates in NSGA-II should be taken as the amplified effect of drift and selection throughout the generations.

Duplicates hinder exploration and selection as well. If duplicates are not eliminated at each generation they accumulate rapidly decreasing the likelihood that the algorithm will explore a larger number of different candidate solutions during a run. Also, since the chances of selecting a given genotype are multiplied by the number of clones of that genotype present in the population, duplicate genotypes end up with higher selective advantage than unique genotypes. This unwanted selective bias is not based in actual fitness and cannot be avoided by ranking procedures, scaling mechanism, or even truncated deterministic mechanisms. A reduced explorative capability combined with an unwanted selective bias can considerably affect the possibility of finding better non-dominated solutions.

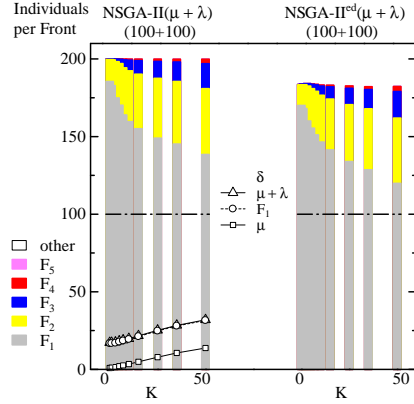
Fig. 4 shows the hypervolume by NSGA-II^{ed}(100+100) that eliminates duplicates and by the conventional NSGA-II(100+100) to illustrate the effect of duplicates on the performance of the algorithms. From this figure, we can see that elimination of duplicates improves the performance of NSGA-II in two and five objective landscapes for $K_i \geq 5$ and $3 \leq K_i \leq 35$, respectively. In three objectives landscapes we see almost no improvement by eliminating fitness duplicates. Note that the largest overall performance difference between NSGA-II^{ed} and NSGA-II is for $M = 2$ objective landscapes, where precisely the accumulation of duplicates in the conventional NSGA-II is the highest as shown in **Fig.**



(a) $M = 2$

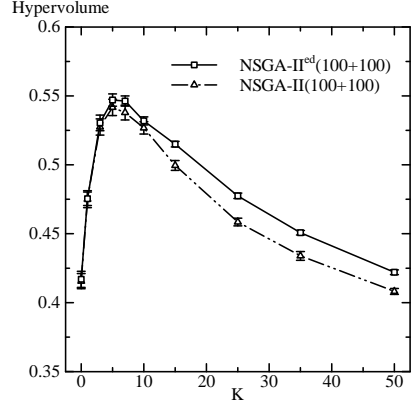


(b) $M = 3$

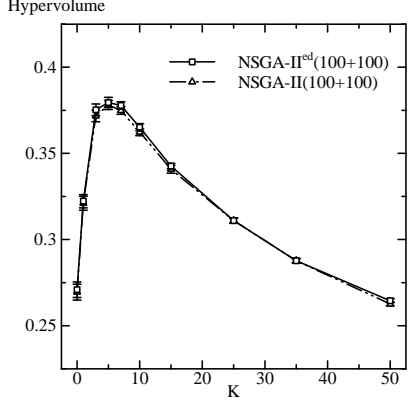


(c) $M = 5$

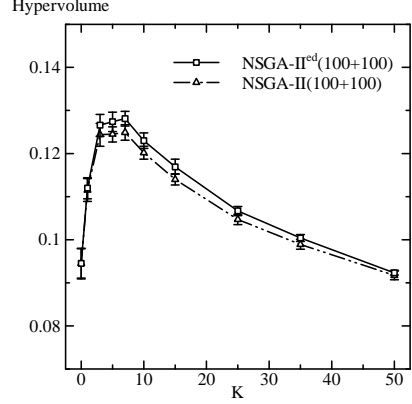
Fig. 3. Average number of individuals per front and average number of duplicates by conventional NSGA-II and enhanced NSGA-II^{ed}. $N = 100$ bits.



(a) $M = 2$



(b) $M = 3$



(c) $M = 5$

Fig. 4. Effect of drift on performance. \mathcal{H} over K by conventional NSGA-II and NSGA-II^{ed} that eliminates fitness duplicates. $N = 100$ bits.

3. Now, since more duplicates are observed for $M = 3$ than $M = 5$ objectives, an interesting question is why do we see almost no improvement for $M = 3$ although we see it for $M = 5$? We answer this question in the next section.

6.3 Selection Bias by Front Level and Objective Space Diversity

The number of epistatic interactions K and the number of objectives M are important factors that determine the density of non-dominated fronts in multiobjective landscapes [6]. First, the number of non-dominated solutions in the top non-dominated fronts reduces as K increases. Second, fixing the bit string length (size of the search space) and increasing the number of objectives M we have landscapes with fewer but more dense non-dominated fronts. Third, the effect on front's density by M is stronger than the effect by K . These properties are clearly reflected in the distribution of solutions per front by NSGA-II^{ed} in **Fig. 3**. Note that for NSGA-II^{ed} the number of individuals in the first front decreases as K increases. However, increasing M the number of individuals in the first front increases and fewer fronts fit in the population.

A consequence of front's density is that it could restrain selection, especially during mating. Taking as an example the way selection is made in NSGA-II, see **3.1**, Pareto non-domination level will be important for mating mostly in $M = 2$ objectives where several fronts fit within the truncated population, except for $K = 0$. Increasing M , fewer but more dense fronts would increase the relative importance of crowding of solutions within a front over non-domination level as criterion to bias selection, especially if most individuals within the truncated population belong to the same front.

From the same **Fig. 3**, we can see that in fact the truncated population of NSGA-II^{ed} would mostly come from the first front for $M = \{3, 5\}$. Thus, in both cases during mating the criterion to bias selection would be mainly crowding factor since most solutions would be ranked with the same non-domination level. However, there is a difference between $M = 3$ and $M = 5$ given by truncation. Truncation would reinforce diversity in objective space by purging individuals with high crowded factor if the number of individuals in the front is larger than the population size. Note that for $M = 3$ in NSGA-II^{ed} the number of individuals in the first front F_1 is close to the size of the truncated population (μ), for most K , and thus the algorithm does not have a chance to purge highly crowded individuals. On the other hand, for $M = 5$ the number of individuals in the first front is greater than the truncated population size (μ), for all K , and truncation can contribute purging highly crowded individuals. This suggests that the increase in performance for $M = 5$ but not for $M = 3$ is because the algorithm by means of truncation is preserving diversity better for $M = 5$.

7 Recombination and Mutation

EAs to function effectively must balance the processes of evolution that decrease genetic variation with those that increase it. In previous sections we have restricted our discussion to selection and drift, which decrease genetic variation.

In the following we focus on the effectiveness of recombination and mutation, mechanisms that increase genetic variation. We also discuss issues that hinder exploration under elitist selection and try ways to make mutation more effective.

7.1 Recombination

In this section we observe the effect on performance of (not) using recombination. **Fig. 5** shows results by NSGA-II^{ed}(100+100) and M^{ed}(100+100). M^{ed} is an NSGA-II^{ed} algorithm with recombination turned off using mutation as the sole variation operator, i.e. $p_c = 0.0$ and $p_m = 1/N$ per bit. From **Fig. 5** note that NSGA-II^{ed} that includes recombination and mutation performs better than M^{ed} that uses only mutation for $K \leq 1$, $K \leq 3$, and $K \leq 7$ for $M = \{2, 3, 5\}$ objectives, respectively. For other values of K we do not see any contribution to performance by including recombination. In fact, we can see that mutation alone performs better for some values of K , especially in landscapes with $M = 2$.

Results by recombination are in accordance with the effects of epistasis on multiobjective landscapes. In [6] it is shown that for small values of K non-dominated solutions of top fronts are highly correlated in decision space (genotype), in objective space (phenotype), and between spaces. However, this correlation decreases rapidly by increasing K , being its fall faster for smaller M . For small K , recombination of high fitness individuals would likely produce high fitness offspring. However, as the number of epistatic interactions K increases the likelihood that offspring would be far from the parents in objective space also increases considerably. The properties of MNK-Landscapes offer no much hope for blind mating and recombination, i.e. just taking any two individuals from the best non-dominated front in the population and recombining them. In the literature there are some reports suggesting that mating based on proximity in decision or objective space could help recombination in MOEAs. It will be interesting to assess in the future the benefit of these approaches on scalable epistatic landscapes. How helpful are them as we increase K ?

7.2 Elite's Age and Mutation Explorative Range

An undesired side effect of elitism combined with a short explorative range by recombination and/or mutation is cyclically exploring same points. To explain this it is useful to see the probability of recombination p_c governing the application of two operators. One is recombination followed by mutation (p_c) and the other one is mutation alone ($1 - p_c$). In this section we focus on elitism and mutation and do not consider the case of recombination followed by mutation.

Conventional NSGA-II, for example, uses a $(\mu + \lambda)$ selection where elite solutions could remain in the population indefinitely. Additionally, mutation rate is often set to $p_m = 1/N$, which means that mutation will explore solutions in average one bit away from the parent in decision space. In this case, eventually after some generations offspring created from elite solutions would likely not be different from offspring created before, even in the case of perfect sampling (no drift). The expected time for mutation to start sampling again same points from

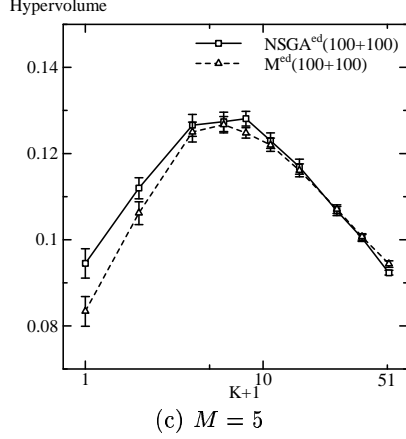
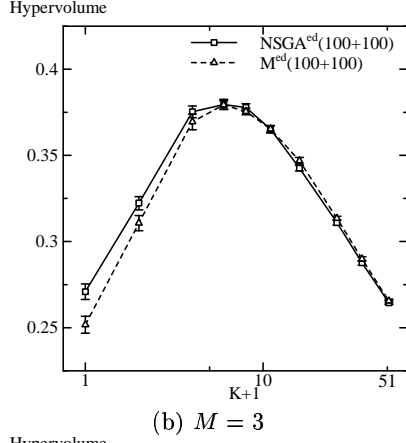
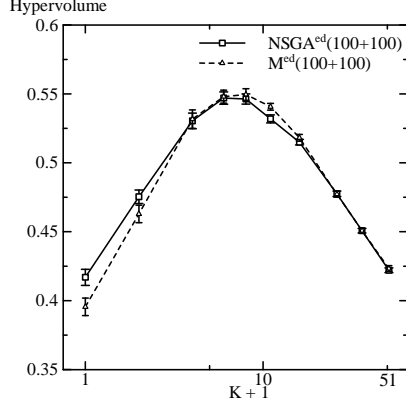


Fig. 5. Effect of recombination on performance. \mathcal{H} over K by NSGA-II^{ed} using crossover followed by mutation and M^{ed} using mutation alone. $N = 100$ bits landscapes.

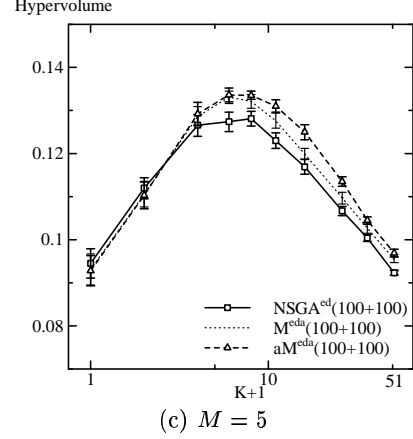
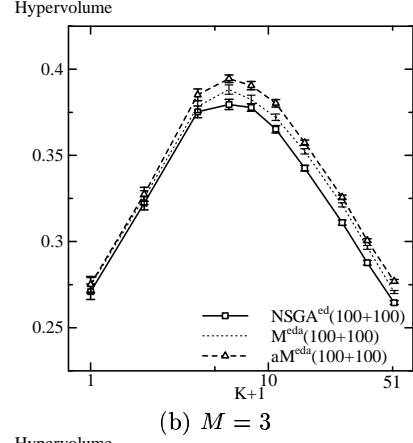
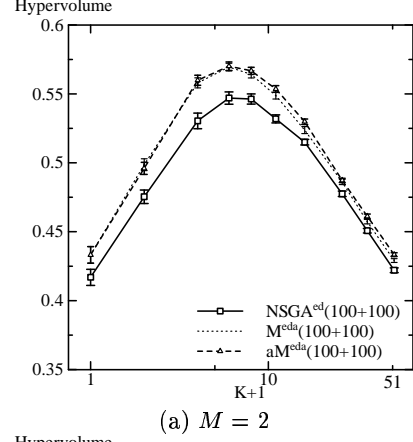


Fig. 6. Elite's age and mutation explorative range. \mathcal{H} over K by NSGA-II^{ed}, M^{eda} that eliminate old elite individuals, and aM^{eda} that also uses age to guide mutation. $N = 100$ bits landscapes.

an elite individual would be a function of the number of occasions the same elite individual has been selected for reproduction, the rate at which mutation is applied alone given by the probability $1 - p_c$, and the bit string length N .

A way to avoid this undesired cycles and enhance exploration is to put an age limit to elite solutions and bias selection accordingly. To observe the effects of age of elite solutions we create M^{eda} from NSGA-II^{ed}. M^{eda} increases by one the age of an elite solution each time it is selected for reproduction. Age is also incremented by one at each generation. Before truncation, fitness duplicates and individuals with age greater than N are eliminated from the population. It uses mutation with probability $p_m = 1/N$ per bit but does not use crossover ($p_c = 0.0$) for simplicity. The rationale for this is that selecting N times the elite individual for reproduction will suffice to sample a good number of solutions that lay within the average explorative range set by the mutation rate, i.e. one-bit neighbors. Of course, mutation with $p_m = 1/N$ will sometimes flip more than one bit, none at all, or sample the same bit more than once during the N trials.

We also verify whether a local search-like strategy would be more effective than the conventional bit flipping mutation strategy. To do that we create aM^{eda} from M^{eda} . aM^{eda} , in addition to eliminating duplicates and very old elite individuals, it also uses the age to guide mutation. The bit string of length N is subdivided in S segments of length L , $N = S \times L$. For elite individuals, age greater than one, mutation flips one bit at the position indicated by $j + i$, $j = S \times [rand() \bmod S]$ and $i = [age \bmod N] \bmod L$, i.e. the mutation segment is chosen at random and the bit within the segments is given by the age of the individual. This kind of mutation makes sure that only one bit will be flipped and increases the chances of exploring most one-bit neighbors of an elite individual as its age approaches N . In our experiments $N = 100$ and $S = L = 10$. For individuals whose age is one standard flipping mutation is applied ($p_c = 0.0$).

Fig. 6 shows results by the mutation-only algorithms $M^{eda}(100+100)$ and $aM^{eda}(100+100)$ together with results by NSGA-II^{ed}. Looking at results by M^{eda} we can see that preventing old elite individuals increases substantially the performance of NSGA-II^{ed} for all number of objectives M and most values of epistatic interactions K . Note that in this case there is no more a performance advantage offered by recombination in small K landscapes, except for $M = 5$ and $K \leq 1$. Looking at results by $aM^{eda}(100+100)$ also note that eliminating old elite individuals combined with local search-like mutation strategy informed by age further improves performance, especially for medium and high K .

8 Comparison with moRBC

Finally, we compare the performance of conventional NSGA-II, the enhanced algorithm aM^{aed} , and the population-based multiobjective random one-bit climber moRBC($\delta : 1 + 1$). Results are shown in **Fig. 7**. From this figure we can see that the performance of conventional NSGA-II is worse by several standard deviations than the performance of moRBC($\delta : 1 + 1$), for all values of K and M . In contrast, note that the performance of aM^{eda} approaches the performance

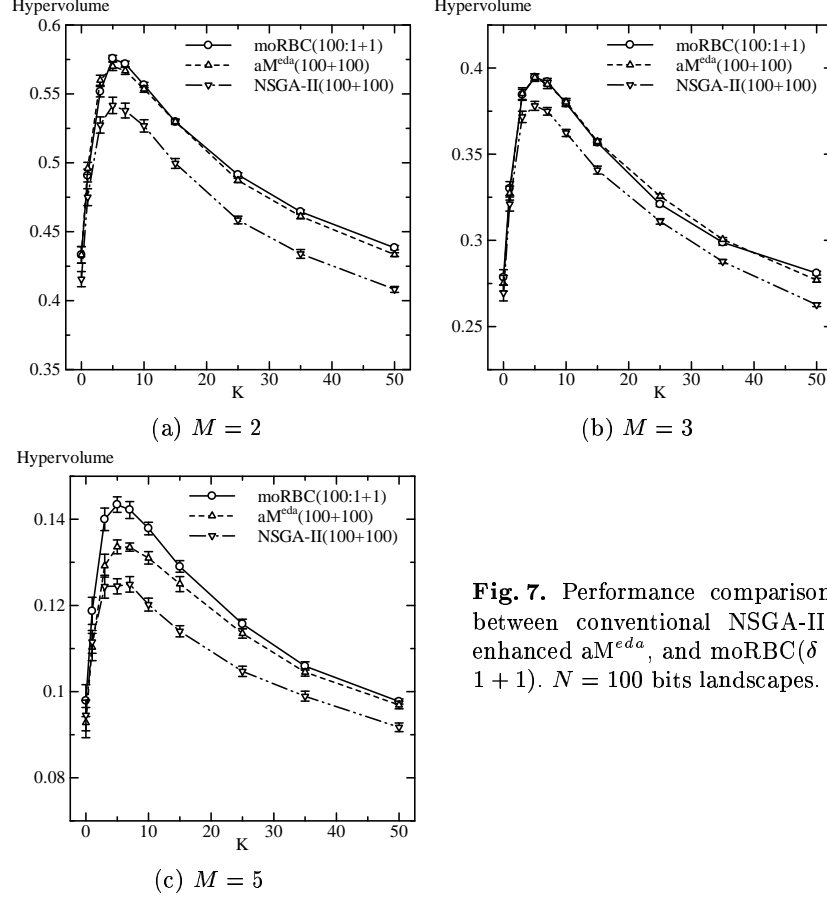


Fig. 7. Performance comparison between conventional NSGA-II, enhanced aM^{eda}, and moRBC($\delta : 1 + 1$). $N = 100$ bits landscapes.

of moRBC($\delta : 1 + 1$) in $M = 2$ and $M = 3$ objectives. However, for $M = 5$ objectives moRBC($\delta : 1 + 1$) still performs better.

9 Conclusions

In this work we have studied the effects of selection, drift, recombination, and mutation in MOEAs on discrete search spaces by using MNK-Landscapes. We have shown that enhancing selection and postponing drift by eliminating fitness duplicates and removing old elite individuals help to increase substantially the performance of MOEAs. We also observed that recombination adds to the performance of standard bit flipping mutation only for small values of epistatic interactions. However, any gain by recombination is largely surpassed by doing a more effective exploration with short-ranged mutation alone. Enhancements in selection, postponing drift, and explorative efficiency have considerably increased the robustness of MOEAs across several classes of epistatic problems and number of objectives. Yet, these enhancements are not enough to surpass the perfor-

mance of simpler population-based multiobjective random one-bit climbers and we should look for ways to design better MOEAs. Results in this work strongly suggest that elitism combined with an efficient short-range explorative capability by mutation is highly effective and likely to be a required feature of MOEAs. The advantages, if any, of elitism combined with mutation using larger explorative ranges should be investigated in the future. In addition, it would be interesting to look into special mating strategies for recombination to further clarify its role in multiobjective discrete search spaces.

References

1. S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, 1993.
2. R. Heckendorn, S. Rana, and D. Whitley, "Test Function Generators as Embedded Landscapes", *Foundations of Genetic Algorithms 5*, pp.183–198, Morgan Kaufmann, 1999.
3. R. Smith and J. Smith, "An Examination of Tunable, Random Search Landscapes", *Foundations of Genetic Algorithms 5*, pp.165–182, Morgan Kaufmann, 1999.
4. K. E. Mathias, L. J. Eshelman, and D. Schaffer, "Niches in NK-landscapes", *Foundations of Genetic Algorithms 6*, pp.27–46, Morgan Kaufmann, 2001.
5. H. Aguirre and K. Tanaka, "A Study on the Behavior of Genetic Algorithms on NK-Landscapes: Effects of Selection, Drift, Mutation, and Recombination", *IEICE Trans. Fundamentals*, (9):2270–2279, 2003.
6. H. Aguirre and K. Tanaka, "Insights on Properties of Multiobjective MNK-Landscapes", *Proc. 2004 IEEE Congress on Evolutionary Computation*, pp.196–203, IEEE Press, 2004.
7. K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimization Test Problems". *Proc. 2002 Congress on Evolutionary Computation*, pp.825–830, IEEE Press, 2002.
8. H. Aguirre and K. Tanaka, "Effects of Elitism and Population Climbing on Multiobjective MNK-Landscapes", *Proc. 2004 IEEE Congress on Evolutionary Computation*, pp.449–456, IEEE Press, 2004.
9. H. Aguirre, M. Sato, and K. Tanaka, "Preliminary Study on the Performance of Multiobjective Evolutionary Algorithms with MNK-Landscapes", *Proc. RISP Intl. Workshop on Nonlinear Circuits and Signal Processing*, pp.315–318, 2004.
10. K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester, West Sussex, England, 2001.
11. E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm", Technical Report 103, TIK-Report, 2001.
12. C. Coello, D. Van Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, Boston, 2002.
13. E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1999.
14. J. Knowles and D. Corne, "On Metrics for Comparing Non-dominated Sets", *Proc. 2002 Congress on Evolutionary Computation*, pp.711–716. IEEE Press, 2002.
15. M. Fleischer, "The Measure of Pareto Optima: Applications to Multi-objective Metaheuristics" *Second Intl. Conf. on Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol.2632, pp.519–533, Springer, 2003.