2004 RISP International Workshop on Nonlinear
Circuit and Signal Processing (NCSP'04)
Hawaii, USA, Mar. 5-7, 2004.

Preliminary Study on the Performance of Multiobjective Evolutionary Algorithms with MNK-Landscapes

Hernán E. Aguirre, Masahiko Sato, and Kiyoshi Tanaka

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

Abstract

Epistasis and NK-Landscapes in the context of multiobjective evolutionary algorithms are almost unexplored subjects. Here we present an extension of Kauffman's NK-Landscapes to multiobjective MNK-Landscapes in order to use them as a benchmark tool and as a mean to understand better the working principles of multiobjective evolutionary algorithms (MOEAs). In this work we present an elitist multiobjective random bit climber (moRBC) and compare its performance with NSGA-II and SPEA2, two elitist state of the art MOEAs.

1. Introduction

Epistasis is a term used in biology to describe a range of non-additive phenomena due to the non-linear interdependence of gene values. In the context of evolutionary algorithms (EAs) this terminology is used to describe nonlinearities in fitness functions due to changes in the values of interacting bits. An implication of epistatic interactions among bits is that the fitness function develops conflicting constraints. That is, a mutation in one bit may improve its own contribution to fitness but may decrease the contributions of other bits with which it interacts. Hence, epistatic interactions increase the difficulty in trying to optimize all bits simultaneously. The influence of epistasis on the performance of single objective EAs is being increasingly investigated. Kauffman's NK-Landscapes model of epistatic interactions [1], particularly, have been the center of several theoretical and empirical studies both for the statistical properties of the generated landscapes and for their *EA-hardness* (see for example [2] and there in). However, the effects of epistasis and NK-Landscapes in the context of multiobjective evolutionary algorithms (MOEAs) are almost unexplored subjects.

In this work we present an extension of Kauffman's NK-Landscapes to multiobjective MNK-Landscapes in order to use them as a benchmark tool and as a means to understand better the working principles of MOEAs. Also, we introduce an elitist multiobjective random bit climber (moRBC) and compare its performance with NSGA-II and SPEA2, two elitist state of the art MOEAs on scalable random epistatic problems.

2. Multiobjective MNK-Landscapes

A multiobjective MNK-Landscape is defined as a vector function $\boldsymbol{f}(\cdot) = (f_1(\cdot),\, f_2(\cdot), \cdots, f_M(\cdot)) : \mathcal{B}^N \rightarrow \Re$ 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. $K = \{K_1, \cdots, K_M\}$ is a set of integers where K_i ($i = 1, 2, \cdots, M$) 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

$$f_i(\boldsymbol{x}) = \frac{1}{N}\sum_{j=1}^{N} f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \cdots, z_{K_i}^{(i,j)}) \qquad (1)$$

where $f_{i,j} : \mathcal{B}^{K_i+1} \rightarrow \Re$ gives the fitness contribution of bit x_j to $f_i(\cdot)$, and $z_1^{(i,j)}, z_2^{(i,j)}, \cdots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with bit x_j in the string $\boldsymbol{x}$. 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 $\boldsymbol{x}$ expressed by a Kauffman's NK-Landscape model of epistatic interactions [1]. **Fig. 1** shows an example of the fitness functions $f_{1,3}$ and $f_{2,3}$ associated to bit x_3 contributing to the first objective function $f_1(\cdot)$ and second one $f_2(\cdot)$, respectively, based on a different epistatic model for each objective.

For a given N, we can tune the ruggedness of the fitness function $f_i(\cdot)$ 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

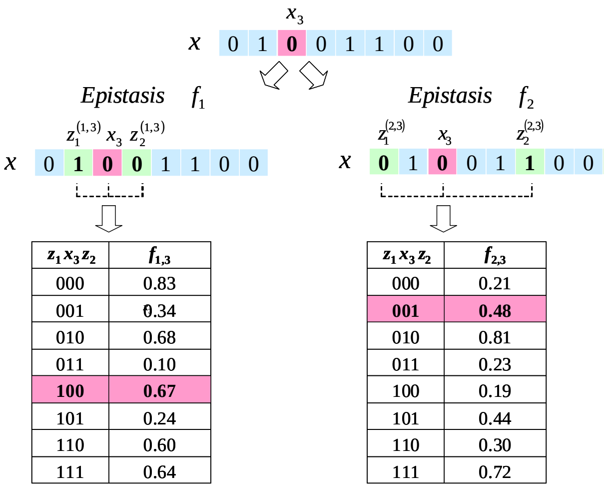

Figure 1: An example of the fitness functions $f_{1,3}(x_3, z_1^{(1,3)}, z_2^{(1,3)})$ and $f_{2,3}(x_3, z_1^{(2,3)}, z_2^{(2,3)})$ associated to bit x_3 contributing to the first objective function $f_1(\cdot)$ and second one $f_2(\cdot)$, respectively.

K_i other interacting bits. That is, the distribution D_i of K_i bits among N. Thus, M, N, $K = \{K_1, K_2, \cdots, K_M\}$, and $D = \{D_1, D_2, \cdots, D_M\}$, completely specify a multiobjective MNK-Landscape and by varying them 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 combinatorial optimization algorithms.

3. Multiobjective Optimization Concepts

Let us consider, without loss of generality, a maximization multiobjective problem with *M* objectives:

$$maximize\ \ \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})) \qquad (2)$$

where $\boldsymbol{x} \in \mathcal{S}$ is a solution vector in the solution space $\mathcal{S}$, and $f_1(\cdot), f_2(\cdot), \cdots, f_M(\cdot)$ the *M* objectives to be maximized. Important concepts used in determining a set of solutions for multiobjective problems are dominance, Pareto non dominated set, Pareto optimal set, Pareto front, and true Pareto front. These concepts are frequently used in this work and can be defined as follows.

Dominance: A solution $\boldsymbol{x}$ dominates other solution $\boldsymbol{y}$ if and only if the two following conditions are true:

1. $\forall m \in \{1, \cdots, M\}\ f_m(\boldsymbol{x}) \geq f_m(\boldsymbol{y})$
2. $\exists m \in \{1, \cdots, M\}\ f_m(\boldsymbol{x}) > f_m(\boldsymbol{y})$.

Dominance of $\boldsymbol{x}$ over $\boldsymbol{y}$ is denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq \boldsymbol{f}(\boldsymbol{y})$. A solution $\boldsymbol{z}$ is nondominated by other solution $\boldsymbol{y}$ if $\boldsymbol{f}(\boldsymbol{y}) \not\succeq \boldsymbol{f}(\boldsymbol{z})$.

Pareto nondominated set: A Pareto nondominated set $\mathcal{P}$ is such that all its elements are nondominated amongst themselves and it is defined by

$$\mathcal{P} = \{\boldsymbol{x} | \neg\exists\ \boldsymbol{y} \in \mathcal{P} : \boldsymbol{f}(\boldsymbol{y}) \succeq \boldsymbol{f}(\boldsymbol{x})\}. \qquad (3)$$

Pareto nondominated optimal set: The Pareto nondominated optimal set is obtained taking into account the whole search space $\mathcal{S}$ and it is defined by

$$\mathcal{P}_1 = \{\boldsymbol{x} \in \mathcal{S} | \neg\exists\ \boldsymbol{y} \in \mathcal{S} : \boldsymbol{f}(\boldsymbol{y}) \succeq \boldsymbol{f}(\boldsymbol{x})\}. \qquad (4)$$

Pareto front: The Pareto front obtained from a nondominated set $\mathcal{P}$ is defined by

$$\mathcal{F} = \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x}))\ |\boldsymbol{x} \in \mathcal{P}\}. \qquad (5)$$

True Pareto front: The true Pareto front is obtained from the Pareto nondominated optimal set $\mathcal{P}_1$ and it is defined by

$$\mathcal{F}_1 = \{\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x}))\ |\boldsymbol{x} \in \mathcal{P}_1\}. \qquad (6)$$

4. The Algorithms

4.1. Random Bit Climber using a Population for Restarts moRBC($\delta : 1 + 1$).

moRBC($\delta : 1 + 1$) begin with a randomly created parent string p of length N. A random permutation of the string positions is generated and a child c is created by flipping one bit of the parent individual following the order indicated by the random permutation. The child substitutes the parent if it fulfills a given REPLACEMENT CRITERION. After testing all N positions indicated by the random permutation, if replacements were detected a new permutation of string positions is generated and testing continues. If no replacements were detected, a local optimum relative to the replacement criterion has been found and the moRBCs opt for a RESTART. This process continues until a given number of evaluations has been expended. The nondominated solutions found during the search are kept in an $Archive$ of a limited capacity. Nondominated individuals with better crowding distance [3](p.236) are preferred by the procedure that updates the $Archive$. Additionally, a *Population* of up to δ solutions nondominated by the parent and amongst themselves are also kept. The procedure used to update the *Population* is the same used to update the *Archive*.

moRBC($\delta : 1+1$) uses dominance as REPLACEMENT CRITERION for climbing. When a *dominance local optimum* has been found, moRBC($\delta : 1 + 1$) RESTARTS the search by replacing the parent with one individual chosen from the collected $Population$. If $Population$ is empty, it performs a hard restart, i.e the parent is replaced by a random string created anew. Since individuals in the $Population$ are nondominated by the parent and also nondominated amongst themselves, restarting the search from the $Population$ implies an elitist strategy. However, some individuals in the $Population$ may actually be dominated by the $Archive$. Hence, this elitism is "local" to the parent. moRBC(δ:1 + 1) incorporate dominance, elitism, and diversity preserving strategies (in objective space), features common to most of the state of the art MOEAs [3, 4].

4.2. NSGA-II

NSGA-II uses an elite-preservation strategy and an explicit diversity-preserving mechanism [3]. 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 from the combined population $\mathcal{R}_t$ of parents and offspring. That is, $\mathcal{R}_t = \mathcal{P}_t \cup \mathcal{Q}_t \wedge \mathcal{P}_{t+1} \subset \mathcal{R}_t$, where $|\mathcal{R}_t| = 2\mu, |\mathcal{P}_{t+1}| = \mu$. To obtain $\mathcal{P}_{t+1}$, $\mathcal{R}_t$ is first classified into nondominated fronts. The first front $\mathcal{F}_1$ contains the best nondominated solutions $\mathcal{S}_1$. The subsequent fronts $\mathcal{F}_j, j > 1$, contain lower level nondominated solutions and are obtained by disregarding solutions corresponding to the previous higher nondominated fronts, ie. $\mathcal{F}_j, j > 1$, is obtained from the set $\mathcal{R}_t - \bigcup_{k=1}^{j-1} \mathcal{S}_k$. Once the classification of nondominated fronts is over, the parent population $\mathcal{P}_{t+1}$ is filled with solutions belonging to the higher fronts, starting with front $\mathcal{F}_1$. Each solution in $\mathcal{P}_t$ is assigned a 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 $\mathcal{Q}_{t+1}$ from $\mathcal{P}_{t+1}$. During selection, a solution i wins a tournament if it has a better rank than j. If i and j have the same rank, the solution with best crowding distance wins.

4.3. SPEA2

SPEA2 [5] creates a population $\mathcal{P}_t$ of μ individuals at the t-th generation, and introduces elitism by explicitly maintaining and external population $\mathcal{E}_t$. $\mathcal{E}_t$ stores a fixed number or the best nondominated solution found since the beginning of the simulation. Each member i of the external population $\mathcal{E}_t$ is assigned a fitness in proportion to the number of individuals n_i in the current population $\mathcal{P}_t$ it dominates. That is, $f_j = \frac{n_i}{\mu+1}$. The fitness of the j-th member of the current population $\mathcal{P}_t$ is calculated taking into account the number of solutions $n_i, i \in \mathcal{E}_t$, that dominate $j \in \mathcal{P}_t$. That is, $f_j = 1 + \sum_{i \in \mathcal{E}_t} f_i$. In other words, solutions in $\mathcal{P}_t$ with lower f_j are better. The next population $\mathcal{P}_{t+1}$ of size μ is created from $\mathcal{E}_t \cup \mathcal{P}_t$ by applying binary tournament selection, recombination, and mutation. SPEA2 uses a nearest neighbor density estimation method with an enhanced truncation method for the external population $\mathcal{E}_t$ in order to efficiently guide the search and to guarantee the preservation of boundary solutions.

5. Metric and Test Problems

In this work we use the hypervolume metric $\mathcal{H}$ proposed by Zitzler [5] to evaluate and compare the performance of the algorithms. Let $\mathcal{A}$ be a set of nondominated solutions. The metric $\mathcal{H}$ calculates the hypervolume of the multidimensional 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

$$\mathcal{H}(\mathcal{A}) = \cup_{i=1}^{|\mathcal{A}|} (\mathcal{V}_i - \cap_{j=1}^{i-1} \mathcal{V}_i \mathcal{V}_j) \qquad (7)$$

where $\mathcal{V}_i$ is the hypervolume rendered by the point $\boldsymbol{x}_i \in \mathcal{A}$ and the reference point. For a given problem, $\mathcal{H}(\mathcal{A}) > \mathcal{H}(\mathcal{B})$ if the elements of $\mathcal{A}$ are closer to the true Pareto front $\mathcal{F}_1$ than the elements of $\mathcal{B}$. The hypervolume is among the few recommended metrics for comparing nondominated sets [6]. Regarding MNK-Landscapes, the hypervolume of the true Pareto front increases with K_i, regardless of N and M. This is an important property to remember in comparative studies of MOEAs.

We conduct our study on MNK-Landscapes with $M = \{2, 3\}$ objectives and $N = 100$ bits, vary the number of epistatic interactions from K_i from $\%0$ to $\%50$ of N simultaneously in all objectives $(K_1, \cdots, K_M = K_i)$, and set *random* epistatic patterns among bits for all objectives $(D_1, \cdots, D_M = random)$. For each combination of M, N and K_i, 50 different problems were randomly generated.

To observe and compare the performance on subclasses of problems we plot the mean value of the *Archive's* hypervolume in the 50 problems over several values of K_i for each value of M. We maximize all objectives and set $[0.0, \cdots, 0.0]$ as the reference point to calculate the hypervolume. Vertical bars overlaying the mean hypervolume curves represent 95% confidence intervals. Throughout this work the number of evaluations is set to 3×10^5 and the *Archive* size is set to 100.

6. Behavior Observation and Discussion

6.1. Effect of population size on moRBC

First, we study the effect of the population size on the moRBC. **Fig. 2** shows results by moRBC($\delta : 1 + 1$) for some values of δ in $M = 2$ and $M = 3$ objectives. From the plots, note that the value of the hypervolume found by the algorithms increases rapidly with K_i, for $K_i = 0$ to small values of K_i. Then, it decreases continuously with K_i, for medium and high K_i. These decreasing values of the hypervolume when K_i increases indicates that the search performance of the algorithms is worsening. Also, it can be seen that the inclusion of a *Population* of size δ for restarts in moRBC($\delta : 1 + 1$) improves noticeably the performance of the bit climber. The best overall performance is achieved by moRBC($100 : 1 + 1$) in two and three objectives. However, note that with small values of δ it is possible to achieve

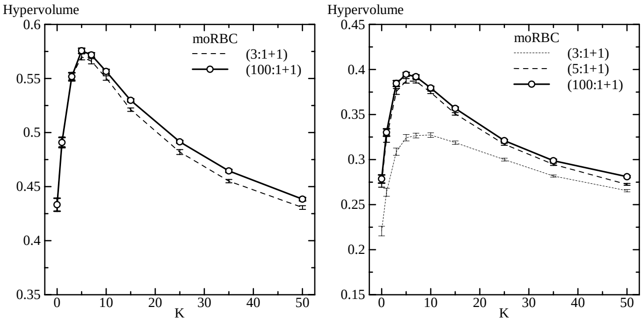

Figure 2: moRBC(δ : $1+1$). $M = 2$ and $M = 3$ from left to right.

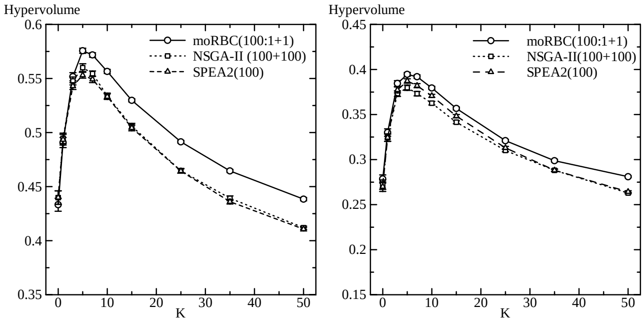

Figure 3: moRBC(δ : $1+1$), NSGA-II, and SPEA2. $M = 2$ and $M = 3$ from left to right

high performance. For example, moRBC(δ : $1 + 1$) with $\delta = 3$, and $\delta = 5$ already approach the performance of moRBC($100 : 1 + 1$) for two, and three objectives, respectively. Note that as the number of objectives increases the size of the population δ also needs to be increased in order to approach the performance of moRBC($100 : 1 + 1$).

To explain better how the population size δ affects the climbing process remember that the population is a set of individuals nondominated by the parent and amongst themselves. Therefore the population establishes a front. Increasing the size of the population would likely increase the coverage of its front, especially if crowding distance is also taken into account to update the population. Thus, the population size increases the quality level (closeness to the true Pareto) by which nondominated solutions are accepted as members of the population, implicitly pushing the algorithm towards higher fronts. Since the population is used for restarts, an algorithm that continues its search from an individual drawn from a larger population is likely to be in a better position to explicitly climb further by dominance to higher fronts.

6.2. Comparison with NSGA-II and SPEA2

Besides results by the moRBC, **Fig. 3** also includes results by NSGA-II and SPEA2. For both MOEAs two-point crossover is used for recombination with probability 0.6, and mutation is implemented as the standard bit-flipping method and applied with probability $1/N$ per bit. The parent and offspring population sizes are set to 100.

From the plots we can see that the overall performance of the multiobjective random bit climber is better than NSGA-II and SPEA2 for two and three objectives. Regarding NSGA-II and SPEA2, note that they perform very similarly for two objectives and most values of K_i. For three objectives SPEA2 performs better than NSGA-II for $5 \leq K_i \leq 15$. NSGA-II and SPEA2 are a population based search heuristic and use elitism. These results raise several questions regarding the different processes involved in NSGA-II and SPEA2. Are selection and recombination performing well? These questions would be the subject of future research.

7. Conclusions

In this work we have presented an extension of Kauffman's NK-Landscapes to multiobjective MNK-Landscapes in order to use them as a benchmark tool and as a means to understand better the working principles of MOEAs. We also introduced an elitist multiobjective random bit climber and compared its performance against NSGA-II and SPEA2. We conducted experiments on MNK-Landscapes with $M = \{2, 3\}$ objectives, $N = 100$ bits, varying the epistatic interactions K from 0 to 50. We observed that moRBC(δ : $1 + 1$) showed superior performance for all values of K_i in $M = 2$ and $M = 3$ objectives. Increasing $_Ki$ the performance of the MOEAs decrease significantly. As future works, we would like to continue studying the effects on performance of the several processes involved in a multiobjective evolutionary algorithm using scalable epistatic problems.

References

[1] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, 1993.

[2] H. Aguirre and K. Tanaka, "Genetic Algorithms on NK-Landscapes: Effects of Selection, Drift, Mutation, and Recombination", *Proc. Third European Workshop on Evolutionary Computation (EvoCOP 2003)*, Lecture Notes in Computer Science, Springer, vol.2611, pp.131–142, 2003.

[3] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, West Sussex, England, 2001.

[4] C. A. C Coello, D. A. V. Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Boston, Kluwer Academic Publishers, 2002.

[5] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*, PhD thesis, Swiss Federal Institute of Technology, Zurich, 1999.

[6] J. Knowles and D. Corne, "On Metrics for Comparing Non-dominated Sets", *Proc. 2002 Congress on Evolutionary Computation*, pp.711–716. IEEE Press, 2002.