Self-Organizing Maps for Multi-Objective Optimization

Dirk Büche **Michele Milano** **Petros Koumoutsakos**
Institute of Computational Science
Swiss Federal Institute of Technology (ETH)
CH - 8092 Zuerich, Switzerland
E-mail: {bueche, milano, petros}@inf.ethz.ch

Abstract

This work introduces novel recombination and mutation operators for multi-objective evolutionary algorithms using self-organizing maps in the context of Pareto optimization. The self-organizing map is actively learning from the evolution path in order to adapt the mutation step size. Standard selection operators can be used in conjunction with these operators.

1 Introduction

Evolutionary Algorithms (EAs) are a standard tool for multi-objective optimization. The appealing property of EAs is the population-based search, which allows approximating the Pareto front in a single optimization run by evolving a population of solutions in a cooperative search towards the Pareto front.
While most of the effort has been used in the development of selection operators and especially fitness assignment techniques for Pareto optimization, the recombination and mutation operators are often copied from standard single-objective algorithms and are non-adaptive over the optimization run. This leads to a lack in performance, since the mutation operator does not exploit any knowledge from prior successful mutations for the adaptation of the mutation step size.
In single objective optimization, knowledge exploitation is prominent and leads to a significant performance gain. The Covariance Matrix Adaptation of Hansen and Ostermeier (2001) embeds information about the path of successful mutations (evolution path) in a covariance matrix.
The CMA cannot be easily transferred to multi-objective optimization, since different solutions in the population converge towards different locations along the Pareto front and not towards a single optimum. Thus we introduce a new adaptation method for the mutation step size in multi-objective optimization using the self-organizing maps (SOM) of Kohonen (2001). The method is inspired by the work of Milano *et al.* (2001), who applied SOM to single objective optimization. The SOM is continuously trained with the current best solutions and thus is tracking the evolution path. The SOM adapts the step size such that it focusses on areas of promising solutions in order to generate an accelerated convergence.

2 Multi-Objective Evolutionary Algorithms

2.1 Overview

The selection operator is often considered the key operator for multi-objective evolutionary algorithms (MOEAs). It consists of the fitness assignment and the selection mechanism itself. The dominance criterion in combination with niching techniques is widely used for the fitness assignment, in order to select in average the less dominated solutions and preserve diversity in the population, respectively. A further substantial element is elitism, a technique of storing always the current best solutions in an archive. For a multi-objective problem, the elite solutions are the nondominated solutions. The archive is then participating in the selection process.
The Nondominating Sorting Genetic Algorithm (NSGA-II) of Deb *et al.* (2000) and the Strength Pareto Evolutionary Algorithm (SPEA2) of Zitzler *et al.* (2001) are two common representatives of MOEAs and implement all previously stated techniques.
These algorithms however, do not describe a mutation or recombination operator. To compare the performance on continuous problems, the authors of SPEA2 and NSGA-II use the polynomial distributed mutation and the simulated binary crossover of Deb *et al.*

(1995). Both methods do not implement any learning process, so they do not exploit any knowledge from the evolution path.

2.2 Self-Organizing Map Multi-objective Evolutionary Algorithms

Self-organizing maps (SOMs) are used to approximate a distribution of points by means of a clustering process. A SOM defines a mapping of an input space $\mathbb{R}^n$ onto a regular lattice of m reference vectors (neurons). A fixed h-dimensional connectivity between the neurons is defined on the lattice. Figure 1 illustrates a SOM with 25 neurons and a two-dimensional quadrilateral lattice, i.e. $n = 2$, $m = 25$, and $h = 2$. A reference vector $w_i \in \mathbb{R}^n$ is associated to each neuron i. The response of the network to an input $x_j \in \mathbb{R}^n$ is defined as the best matching neuron c:

$$c(x_j) = \arg\min_i \{||x_j - w_i||\} \quad (1)$$

After the neuron c has been determined, all SOM neurons are *updated* so as to become closer to the input x_j (Kohonen, 2001):

$$w_i^{new} = w_i^{old} + \alpha H(c, w_i) \cdot (x_j - w_i), \quad i = 1, ..., m \quad (2)$$

where $\alpha \in]0, 1[$ is the learing rate and $H(c, w_i)$ is the so-called *neighborhood function*, defined so as $H(c, c) = 1$, $H(c, w_i) \geq 0 \quad \forall w_i$, and its value decreases with the distance between c and w_i as measured on the h-dimensional topology (Kohonen, 2001). The neighborhood function allows the SOM to approximate a given distribution in an ordered fashion, by preserving neighborhood relationships.
Here we approximate the Pareto front with a SOM, by modifying its training algorithm. To this aim we set the SOM connectivity h of the lattice to one dimension less than the objective space, that is the same dimension of the Pareto front; also, since the SOM is defined in design space, the dimension n of its reference vectors is equal to the number of design variables. The SOM is trained by the update procedure on the current parent population of the optimization process in order to approximate the parent population in an ordered fashion. Any *selection operator* like SPEA2 or NSGA-II can select the parent population.
A common *recombination operator* is the intermediate recombination, which computes the design variables of a new solution $u \in \mathbb{R}^n$ by recombining two parents $a, b \in \mathbb{R}^n$:

$$u_k = \beta a_k + (1 - \beta)\, b_k \ , \ k = 1...n, \quad (3)$$

where β is a uniform random number within $[0, 1]$. We define a recombination operator by using the SOM. The SOM is trained on the design variables of the parent population. Thus choosing a random point within the area that is covered by the SOM represents an intermediate recombination of the parent population. The recombination procedure starts by randomly choosing a simplex of adjacent neurons in the lattice, and finally the random point is obtained from a uniform probability distribution within the simplex (Figure 1).
In addition, a *mutation operator* is defined in order to generate points outside the area covered by the SOM. Normally distributed random numbers are added to the new point u by:

$$u_k \leftarrow u_k + \frac{\sigma}{\sqrt{n}} \mathrm{N}(0, 1), \quad k = 1...n, \quad (4)$$

where σ is the step size and is set equal to the Euclidean length of a randomly chosen edge of the simplex. This step size definition includes a step size adaptation. At the beginning of the optimization run, the SOM is initialized with random values and thus large distances between neighboring neurons in the lattice. The average distance of the neighboring neurons is decreasing over the optimization run, since the SOM is aligning along the Pareto front in an ordered fashion and the step size is decreasing.

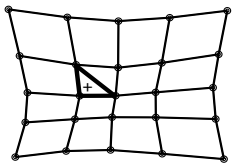

Figure 1: SOM with 25 neurons [circles] and 2D quadrilateral lattice [thin lines]. A random simplex of adjacent neurons is created [bold line]. Within the simplex a uniformly distributed random point [plus symbol] is generated.

3 Experimental Results

The performance of the SOM-MOEA is analyzed for the two-objective test function of Fonseca and Fleming (1995):

$$f_{1/2} = 1 - \exp\left(-\sum_{i=1}^{n}\left(x_k \pm \sqrt{1/n}\right)^2\right) \quad (5)$$

with $x_{1...n} \in [-1, 1]$. The exact Pareto front can be derived as:

$$f_{1/2} = 1 - \exp\left(-n\left(t \pm \sqrt{1/n}\right)^2\right)$$

with $x_{1...n} = t$, $-\sqrt{1/n} \leq t \leq \sqrt{1/n}$. The number of design variables is set to $n = 10$. An optimization run is started with the SOM-MOEA and a population size of 60 individuals. A simple selection operator was generated by selecting only the current nondominated solutions in an elitistic fashion. In order to keep diversity within the selected set, the clustering algorithm of SPEA2 is used, allowing a maximum number of 30 nondominated solutions.
A one-dimensional SOM was initialized with 20 neurons, and random values for the reference vectors. After each generation, the SOM is trained with 30 training steps on the current parent population. The initial population was randomly generated. Figure 2 shows the initial population and SOM for two dimensions of the design space and for the objective space. Consider that a simplex for this SOM is a straight line.
The optimization run was started computing in total 3.000 solutions. The final population is shown in Figure 3. The figure shows that the SOM is aligned along the analytical Pareto front in design space, and the objective values of the final population are well distributed along the Pareto front. The step size of the normally distributed mutation in Equation 4 is related to the length of a simplex, i.e. for this one-dimensional network it is equal to the distance between two adjacent neurons. The ratio of the initial and final step size is equal to the distance between adjacent neurons of the SOM in Figure 2 and Figure 3.

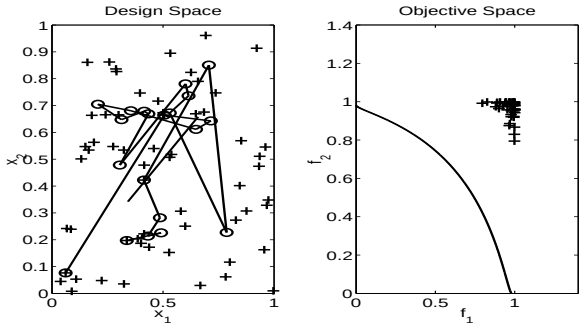

Figure 2: Intitialization of the SOM-MOEA for the two-objective problem of Fonseca and Fleming with 10 design variables: Random population [crosses], random 1-dim. SOM [connected circles] and analytical Pareto front [line] in a 2-dim subspace of the design space and in the objective space.

4 Conclusions

In this report, a self-organizing network is introduced for the first time in a multi-objective optimization con-

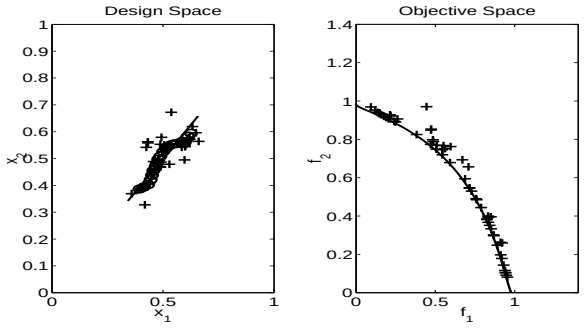

Figure 3: SOM-MOEA after 50 generations (3000 evaluated solutions). The SOM alignes in the design space along the Pareto front.

text. The network comprises the principle of cooperative search by interpolating the current nondominated front, thus it is sharing information about successful design variables values along the Pareto front. In addition, the network describes a novel principle for the adaptation of the step size for the mutation operator. The step size is related to the distance of neighboring neurons in the network and thus varies along the network. This allows different adaptation for different areas along the Pareto front.

Acknowledgments

The first author would like to acknowledge support from the KTI Project 4817.1 and Alstom (Switzerland) AG.

References

K. Deb, and R. B. Agrawal (1995). Simulated binary crossover for continuous search space. *Complex Systems*, No. 9, pp. 115-148.

K. Deb, A. Pratap and T. Meyarivan (2001). Constrained Test Problems for Multi-objective Evolutionary Algorithms. In E. Zitzler et al. (eds.), *Lecture Notes on Computer Science*, Vol. 1993, Springer Verlag.

K. Deb, S. Agrawal, A. Pratap and T. Meyarivan (2000). A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pp. 849-858.

M. C. Fonseca and P. J. Fleming (1995). Multiobjective genetic algorithms made easy: Selection, sharing and mating restrictions. *Proceedings of the 1st International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*,

pp. 45-52, London, UK.

N. Hansen and A. Ostermeier (2001). Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, Vol. 9, No. 2, pp. 159-195.

T. Kohonen (2001). *Self-organizing maps. Springer series in information sciences*, 3rd ed.

M. Milano, J. Schmidhuber, P. Koumotsakos (2001). Active Learning with Adaptive Grids. *International Conference on Artificial Neural Networks*, Vienna, Austria.

E. Zitzler, M. Laumanns, and L. Thiele (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *EUROGEN 2001*, Athens, Greece.