Active Learning with Adaptive Grids

Michele Milano[1], Jürgen Schmidhuber[2], and Petros Koumoutsakos[1⋆]

[1] Institute for Computational Sciences, ETH Zürich, Switzerland
{milano,petros}@inf.ethz.ch
[2] IDSIA, Galleria 2 Manno, CH-6928, Switzerland juergen@idsia.ch

Abstract. Given some optimization problem and a series of typically expensive trials of solution candidates taken from a search space, how can we efficiently select the next candidate? We address this fundamental problem using adaptive grids inspired by Kohonen's self-organizing map. Initially the grid divides the search space into equal simplexes. To select a candidate we uniform randomly first select a simplex, then a point within the simplex. Grid nodes are attracted by candidates that lead to improved evaluations. This quickly biases the active data selection process towards promising regions, without loss of ability to deal with "surprising" global optima in other areas. On standard benchmark functions the technique performs more reliably than the widely used covariance matrix adaptation evolution strategy.

1 Introduction

A central problem of optimization is to select promising solution candidates without wasting too much time on others. This problem of efficient active data selection is an essential motivation of a wide variety of optimization techniques including genetic algorithms, evolution strategies, reinforcement learning algorithms, tabu search, etc.

To illustrate the problem we focus on stochastic optimization techniques such as evolution strategies (ES) [2], which are standard tools used for the optimization of multivariate, possibly non-continuous functions.

Given an n-dimensional search space, ES uses a population of points (n-dimensional "parents") to generate offspring at random from gaussian distributions with the parents as mean, and a standard deviation (step size) that is continually adapted based on information about past successes. This so-called *mutation* process provides ES with the ability to escape from local minima.

To improve efficiency in terms of convergence speed, however, adaptation of the step size during the optimization process is of crucial importance [2] [3]. The set of all mutation steps yielding improvements is called the *evolution path* of the ES [4]. Clearly, it makes sense to exploit the information embedded in the evolution path to accelerate convergence.

A widely used technique called *covariance matrix adaptation evolution strategy* (CMA-ES) embeds the information about the evolution path in a covariance

⋆ also with CTR, NASA Ames 202A-1, Moffett Field, California 94035

matrix describing correlations between previous successful mutation steps. Subsequent mutation steps are forced to be uncorrelated with previous ones, thus optimizing step size.

One drawback of this approach is that information is acquired by means of a *local* process which does not improve the global performance properties of the ES [4].

Here we use an adaptive grid or self-organizing map (SOM [1]) to trace the evolution path. We define an SOM-based mutation operator to generate new offspring; the SOM is continuously trained on successful offsprings only, thus reflecting the evolution path in a way that allows for selection of further successful candidates with high probability. We will see that the information collected by the SOM during the optimization process is global rather than local, thus overcoming problems of CMA-ES.

Section 2 will describe the new method in detail; Section 3 will briefly review CMA-ES and present results on benchmark functions for both methods; Section 4 will conclude and provide an outlook.

2 The SOM-ES algorithm

We start by introducing relevant definitions.

Definition 1. *$f(.)$ is the scalar valued objective function to be optimized*

Definition 2. *$\boldsymbol{x} \in \mathbb{R}^n$ denotes a parameter vector in which $f(.)$ is evaluated*

Definition 3. *$\boldsymbol{x}_{best}$ is a time-varying, variable parameter denoting the $\boldsymbol{x}$ corresponding to the best currently known $f(\boldsymbol{x})$*

Definition 4. *$F_{best} = f(\boldsymbol{x}_{best})$; T is a threshold value for F_{best} used to decide if satisfactory convergence has been attained*

Definition 5. *$\boldsymbol{X}_i$, $i \in [1, \ldots, m]$ are the m SOM codebook vectors or nodes of the adaptive grid*

The SOM has n-dimensional connectivity. Its nodes or codebook vectors partition the search space into simplexes; nodes on the search space boundary are fixed. In Fig.1 a subdivision for $n = 2, m = 36$ is shown.

We are now ready to define a mutation operator $\mathbf{M}$ as follows:

1. Select a simplex at random in the SOM
2. generate a random point $\boldsymbol{x}$ uniformly distributed within the simplex.

Given an optimization problem, the SOM-ES algorithm can be outlined as follows:

1. Initialization:
   (a) distribute the codebook vectors such that they are roughly equally spaced in the search volume

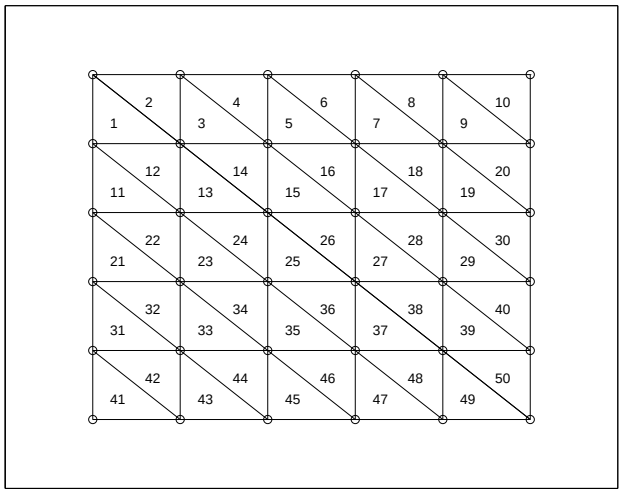

Fig. 1. Subdivision of a 2-dimensional SOM into simplexes. Circles denote codebook vectors; numbered triangles denote simplexes.

(b) Use $\mathbf{M}$ to generate a first point $\boldsymbol{x}_{best}$, and set $F_{best} = f(\boldsymbol{x}_{best})$

2. Use $\mathbf{M}$ to generate a new $\boldsymbol{x}$ and compute $f(\boldsymbol{x})$
3. if $f(\boldsymbol{x}) < F_{best}$ then:
   (a) $\boldsymbol{x}_{best} = \boldsymbol{x}$; $F_{best} = f(\boldsymbol{x})$
   (b) perform a Kohonen node adaptation step [1] on the SOM with $\boldsymbol{x}_{best}$ as input, but restrict nodes on the boundary to move on the boundary only (in the experiments we actually fix the boundary nodes).
4. if $F_{best} > T$ then go to 2.

The Kohonen training step is defined in the standard way:

$$\boldsymbol{X}_i^{new} = \boldsymbol{X}_i + \eta \cdot h(i,w) \cdot (\boldsymbol{x} - \boldsymbol{X}_i), \quad i \in [1,\ldots,m] \tag{1}$$

where w is the index of the codebook vector nearest to $\boldsymbol{x}$ (the "winning" vector); $h(i,w)$ is the neighbourhood function; η is the learning rate. Without step 3b, the SOM-ES algorithm degenerates into pure random search (approximated by the initial optimization stages). Since the SOM is trained only on offspring yielding an improvement of F_{best}, however, allocation of successive offspring through subsequent mutations tends to focus on points in the vicinity of the most successful points encountered so far. This feedback mechanism usually yields a natural evolution path without many unnecessary deviations, as will be illustrated on the standard test functions in the following section. But since the SOM-defined simplexes always cover the entire search space, there always will be a nonvanishing probability of selecting points other than those near successful previous

points — this prevents the method from getting stuck in local optima, and makes it a *global* method.

3 Experimental Comparison with CMA-ES

A widely used optimization algorithm that exploits information conveyed by sequences of successful mutations is the *covariance matrix adaptation evolution strategy* (CMA-ES) [4]. Its fundamental mutation operator for generating new offspring is:

$$\boldsymbol{x}^{new} = \boldsymbol{x}^{old} + \delta \cdot \boldsymbol{B} \cdot \boldsymbol{z} \tag{2}$$

where δ is a global step size; $\boldsymbol{z} \sim N(\boldsymbol{0}; \boldsymbol{I})$ is a random vector drawn from a normal distribution; $\boldsymbol{B}$ is the set of eigenvectors of the covariance matrix of the distribution of successful mutation points. The matrix $\boldsymbol{B}$ may be viewed as a rotation matrix that allows for optimizing the direction in which to generate a new point; all information needed for the calculation of $\boldsymbol{B}$ is gathered from points generated during the optimization process itself. The step size δ is also adaptive — due to lack of space we refer the reader to [4] for full details.

To analyze the performance of SOM-ES we consider three 2-dimensional test functions:

1. Modified Rosenbrock function:

$$f(x,y) = 74 + 100 \cdot \left(y - x^2\right)^2 + (1-x)^2 - 400 * e^{-\left(\frac{(x+1)^2+(y+1)^2}{0.1}\right)} \tag{3}$$

$$(x,y) \in [-2,2] \times [-2,2]$$

2. Griewangk's function:

$$f(x,y) = 1 + \frac{1}{200}\left(x^2 + y^2\right) - cos(x) \cdot cos\left(\frac{y}{\sqrt{(2)}}\right) \tag{4}$$

$$(x,y) \in [-100,100] \times [-100,100]$$

3. Rastrigin's function:

$$f(x,y) = 20 + \left(x^2 - 10 \cdot cos(2\pi x)\right) + \left(y^2 - 10 \cdot cos(2\pi y)\right) \tag{5}$$

$$(x,y) \in [-5.12,5.12] \times [-5.12,5.12]$$

The global minimum of test functions 2 and 3 is $(0,0)$; function 1 is classical Rosenbrock with minimum in $(1,1)$ plus a gaussian bump in $(-1,-1)$. This modification causes a local minimum in $(1,1)$ and a global minimum in $(-1,-1)$, which makes function 1 difficult to optimize, because the local minimum basin is larger than the global mimimum basin. Therefore an algorithm exploiting only local information is very likely to be driven into the local minimum.

In all tests we used two SOMs, with grids of 5×5 and 7×7 codebook vectors, respectively. The learning parameter was always fixed to $\eta = 0.2$. Data points passed to the SOM adjuster were normalized between $[-1, 1]$.

In our initial study the SOM parameters were kept constant during the optimization process, in contrast to what is usually done by the more sophisticated standard training algorithm [1]. The possibility of implementing an adaptive parameter adjustment process will be subject of further studies.

Figures 2-4 trace the evolution of the grid used by 7×7 SOM-ES. We observe that the SOM codebook vectors quickly focus the search on areas that contain the most promising points.

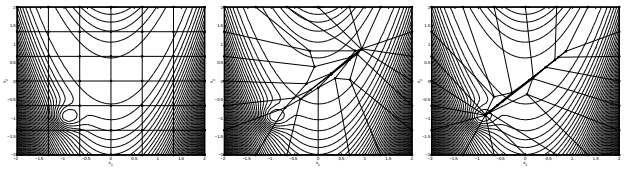

Fig. 2. Evolving SOM superimposed on contour plots of function 1. From left to right: situation after initialization, after 60 iterations, and after 100 iterations. Only neighbourhood connections are shown.

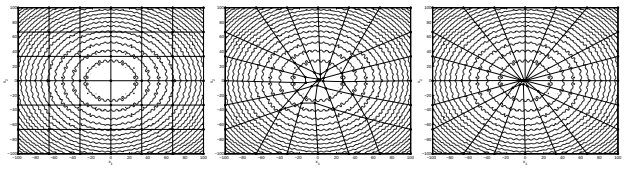

Fig. 3. Evolving SOM superimposed on contour plots of function 2. From left to right: situation after initialization, after 300 iterations and after 700 iterations.

To perform a fair comparison with CMA-ES we used the same convergence criterion for both methods. The thresholds for satisfactory convergence were fixed to 40 for function 1, and to 10^{-3} for functions 2 and 3. There was a maximum of 5000 iterations per run.

To obtain statistically significant results we performed a total of 10000 optimization runs per test function and method, each run with a different initial point for CMA-ES (unnecessary for SOM-ES). To alleviate initialization dependence and convergence problems of CMA-ES, we also evaluated a CMA-ES variant

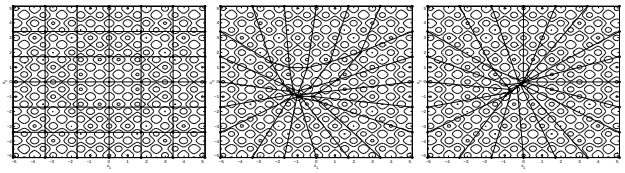

Fig. 4. Evolving SOM superimposed on contour plots of function 3. From left to right: situation after initialization, after 100 iterations and after 200 iterations.

that reinitializes and restarts CMA-ES whenever it fails to converge within 500 iterations. The number 500 actually was chosen with hindsight ("cheating" in favor of CMA-ES), as it is twice the maximum average number of iterations required for CMA-ES to converge on the test functions (compare Table 1).

The results in table 1 demonstrate that SOM-ES consistently outperforms CMA-ES in terms of convergence frequency. On the other hand, the average number of iterations required to achieve convergence is smaller for CMA-ES. This is not surprising as CMA-ES is a local method devised for optimal exploitation of local information. CMA-ES's local focus also is evident from the fact that the restart variant of CMA-ES exhibits improved convergence frequency. Its average number of required iterations tends to become comparable with the one of SOM-ES; SOM-ES's convergence frequency is still better though.

The advantages of SOM-ES are most evident in case of the most difficult function 1 whose difficulty stems from the additional gaussian bump containing the global minimum: the local minimum basin is much larger than the global minimum basin, and the global minimum is not centered in the search domain (a centered global optimum with large basin facilitates the task of many optimization algorithms [5]).

We expect that the harder the optimization problem the more pronounced the advantages of SOM-ES — therefore we have started to apply the method to much more complex optimization problems from the field of computational fluid dynamics, to be reported elsewhere. But even on the easier test functions SOM-ES achieves convergence on global minima much more frequently than both CMA-ES and the restart variant of CMA-ES.

Table 1. Performance comparison: SOM-ES vs CMA-ES on 3 test functions

	Test 1		Test 2		Test 3	
Method	Av. Iter.	% Succ.	Av. Iter.	% Succ.	Av. Iter.	% Succ.
5×5 SOM-ES	130	93%	1600	79%	180	94%
7×7 SOM-ES	100	97%	750	94%	200	97%
CMA-ES	50	24%	340	28%	100	61%
CMA-ES with restarts	80	41%	580	67%	140	89%

4 Conclusions and Outlook

We proposed a conceptually simple yet novel approach to active data selection. It is based on adaptive grids or self-organizing maps trained to reflect relevant structure of an error or fitness landscape reveiled by successful candidates encountered during an optimization process.

Although the method tends to focus computational resources on the neighborhood of previously observed successful candidates, it never loses the ability to discover global optima. Unlike widely used CMA-ES, the novel algorithm does not require random initialization. On standard test functions it consistently outperforms CMA-ES in terms of reliability, without necessity for parameter tuning.

We believe the principles put forward here are quite general and suggest a number of closely related promising algorithms. One of the most general approaches that we are currently evaluating experimentally is this:

Given an n-dimensional confined search space S, define m initially equally spaced, variable grid points $\in S$, *without* any prewired topology. The set of grid points P includes a subset $B \subset P$ of points that may move only on the boundary of S. Like in the present paper, points in P are attracted by solution candidates yielding improved performance on some optimization problem, where the movements of points in B are restricted to the boundary of S. Each time we select a new solution candidate, we simply perform an n-dimensional Voronoi tesselation of the entire search space based on the current positions of all points in P, then uniform randomly select one of the simplexes in the search space partition, then a point within the selected simplex. Preliminary experiments with this topology-free approach already led to excellent results; a detailed analysis will be published elsewhere.

References

1. T. Kohonen, *Self-Organizing maps*, Springer Verlag 1995
2. H. P. Schwefel, *Evolution and Optimum Seeking*,Wiley 1995
3. T. Bäck, U. Hammel, H. P. Schwefel, "Evolutionary Computation. Comments on the History and Current State", IEEE Trans. on Evolutionary Computation, vol. 1, n. 1, 1997, pp. 3-17
4. N. Hansen, A. Ostermeier, "Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation", IEEE Intern. Conf. on Evolutionary Computation (ICEC) Proceedings, 1996, pp. 312-317
5. D. Whitley, K. Mathias, S. Rana, J. Dzubera, "Building Better Test Functions", Proc. of the 6th Int. Conf. on GAs, Morgan Kaufmann, 1995, pp. 239-246