REPORT OF CMA-ES

BY PHILIPP KNESTEL

Contents

Date: December 10, 2019.

1. Abstract

This report covers the paper Completely Derandomized Self-Adaptation in Evolution Strategies (CMA-ES) by Nikolaus Hansen and Andreas Ostermeier. The paper presents two methods of self-adapting the mutation distribution - the concepts of derandomization and cumulation. The goal here is to produce a completely derandomized self-adaptation scheme that adapts any normal mutation distributions. This scheme, called Covariance Matrix Adaptation (CMA), is achieved by targeting the mutation strategy parameters to favor previously selected mutation steps in the future.
To improve the result, even more, use an evolution path instead of individual search steps. The paper then compares local and global search properties of the evolutionary strategy with and without covariance adjustment. The methods and experiments discussed in the paper are presented and explained in this report. It turns out that the services are comparable only with perfectly scaled functions. However, since the application areas for these schemes are complex problems without a known perfect solution, this is an absolute advantage of CMA-ES.
This report discusses the background of the paper as well as related work in this field of research. Finally, the report discusses the effectiveness and impact of the paper on future work.

2. Introduction

The Evolutionary Strategy (ES) follows the basic idea of biological evolution at the stochastic level. We assume complex problems such as optimal market strategies or complex search strategies. The desired goal for the first problem - perfect market prediction for maximum profit - is obvious, but an optimal solution to this problem is not in sight. The goal of ES for problems of this kind is therefore to find the best possible solution for us. Which is usually just a better solution than the previous solution to the problem. The following search problem should be solved: "Minimize a nonlinear objective function, which is a mapping from the search space to the target." (Hansen and Ostermeier, o. J., p. 1).
The search is done step by step, which proceeds by stochastic variation (called mutation in the following) of already found recombination points[1]. From this, you get after each combination/search step new points of this kind from which in turn selects the best options. To perform a mutation, one adds a realization of a normally distributed random vector. The parameters of the normal distribution in which these random vectors move are therefore of paramount importance. These parameters ultimately decide how strong the mutation should be. This paper is concerned with just this set of parameters to optimize the search function. These parameters are referred to below as strategy parameters.
1

[1]Hansen, N. H., and Ostermeier, A. O. (o. J.). Completely Derandomized Self-Adaptation in Evolution Strategies. Abgerufen 30. Oktober 2019, von http://www.cmap.polytechnique.fr/ nikolaus.hansen/cmaartic.pdf

The real difficulty, however, is that you use ES for complex problems. The appropriate choice of the requirements of the strategy parameters is, therefore, the actual topic of this paper. However, these requirements can also be variable and change during the search process. "For this reason, the self-adaptation of the mutation distribution that dynamically adapts strategy parameters during the search process is an essential feature in ESs" (Hansen and Ostermeier, o. J., p. 2) "The mutation step from generation g to g+1 reads for each offspring k = 1, ... ,λ

(1) $x_k^{g+1} = x^{(g)} + \sigma_k^{(g+1)} z_k$

(2)$\sigma_k^{g+1} = \sigma^{(g)} exp(\xi_k)$

where: ξ_k in set of real numbers for $k = 1, ..., \lambda$ independent realizations of a random number with zero mean. Typically, ξ_k is normally distributed with standard deviation $1/\sqrt{2n}$ (Baeck and Schwefel, 1993). We usually prefer to choose $P(\xi_k = 0.3) = P(\xi_k = -0.3) = 1/2$ (Rechenberg, 1994).
And "Components of z_k and identically (0,1)-normally distributed" (Hansen and Ostermeier, o. J., p. 3). As you can see the standard deviation of ξ_k influences the spread of the next searching step. This represents the mutation strength of the strategy parameter level. These steps repeat with each selected strategy parameter. As a result, better results are achieved on an ongoing basis, and the ultimate goal of favoring strategy parameter setting over others is thus achieved. The paper builds on these same developments to come in the later course on CMA, the covariance matrix adaptation. This covariance adjustment in conjunction with ES is currently a highly promising solution to complex variable problems without a known perfect solution.

3. Related Work

The basis for this paper is, on the one hand, the development of Reifenberg 1973. The paper is based on this work in terms of experience with the breadth of the strategy parameter settings. Here it was found that there is only a small range of strategy parameter settings for which considerable search progress can be observed (Reichenberg 1973).
Recent precursors of this work are "An Overview of Evolutionary Algorithms for Parameter Optimization: Evolutionary Computation" by Baeck, T. and Schwefel, H.-P. (1993) and the book "Evolution Strategy '94" by Ingo Rechenberg (1994). These two works each specified the breadth of the strategy parameter settings with other solutions. Whereby the variance represents the mutation level on the respective strategy parameter level.
From the two authors of this paper itself also comes a direct predecessor of this work. The elaboration "step-size adaptation based on non-local use of selection information." In Davidor, Y. et al., Editors, Proceedings of PPSN IV, Parallel Issue Solving from Nature, pages 189-198, Springer, Berlin, Germany." introduced the first stage of derandomization into the control of strategy parameters. Thus, an individual adaptation of the step size was possible in constantly small populations. However, this population size must be scaled linearly to the problem dimension n to accommodate the individual step sizes.

Also very interesting are works that have applied this technique in the industrial environment and report on the results and the actual benefit. It is the 7th Interim Report of the Department of Bionics and Evolution Technology of the Technical University Berlin under grant 01 IB 404 A of the Federal Minister for Education, Science, Research and Technology by Evotech (1997). In this work, one can read the first successful investigations on the use of an evolutionary strategy for time series analysis and observe the developments in the further course. This work is also based on the work of Mr. (Ostermeier 1997) CMA-ES is also used in research fields such as neuro-computing. So you can see significant progress here too. In the work "Optimization of neural fields models: Neurocomputing", 36 (1-4): 225-233 by Igel, C., Erlhagen, W., and Jancke, D. (2001) one can see how CMA-ES in the Compared to other optimization methods.

Nikolaus Hansen and Andreas Ostermeier have already researched this topic in numerous projects and have continuously improved the approach to optimization through evolutionary strategies and ultimately also through CMA-ES. Her work is currently laying the foundation for the accomplishments achieved in this paper. A current dissertation by Johannes Michael Sundberg on "Process Optimization of Industrial Methanol Synthesis Using Gradient-Based and Derivation-Free Optimization Algorithms" shows applications in process optimization in engineering (Sundberg, 2019).
[2]

[2]Sundberg, J. M. S. (2019, Januar 15). Mediatum [PDF]. Abgerufen 11. Dezember 2019, von https://mediatum.ub.tum.de/doc/1470185/1470185.pdf

4. Background

The evolutionary strategy in computer science sees its role model in biological evolution and attempts to copy the process directly. The biological processes to overcome the complex problem of the Earth's ecosystem are called evolution. First recorded by Charles Darwin in the 19th century. Computer science seeks to idealize and simplify the idea behind evolution so that evolutionary strategies can be applied as widely as possible and remain viable. Fundamentally, evolution consists of three steps:
Selection (survival of the strongest) Mutation (spontaneous change of DNA) Combination (sexual reproduction)
These are exactly the steps that CMA-ES or general evolutionary strategies try to imitate. The algorithm should select the best possible options and continue working with them. Afterward, it is tried to change these already good options by mutations and to combine them with other changed options. The mutation strength is decisive here. Because as in the evolution is trying to keep the variability initially high and then to specialize the best solutions. So, initially, you can cover a wide range of solutions and, once you've found the good options, can improve the algorithm to that effect.

First evolutionary strategies were developed in the 60s. However, these could not be implemented due to the low computing power. Nevertheless, the first genetic algorithms were developed at that time and evolutionary programming originated. Different directions of the evolution strategies developed which are however now merged again. However, for the time being, a standard evolutionary algorithm has evolved: initialize the data and the problem, and evaluate the initial options that the algorithm has. This is followed by a first pair selection and recombination. The new options or points let you mutate and evaluate you again. Better than in biological evolution one can draw direct comparisons in computer science and concretely select the better options based on desired parameters. The original idea is now followed by environmental selection. It is therefore tested beyond the desired parameters beyond whether the new mutated combinations meet the requirements. Subsequently, termination conditions are checked. In the last step, it is usually difficult to determine when exactly the algorithm should stop at all. The problems are usually very complex and therefore difficult to assess. So it can happen that the limits are too ambitious or too small.
[3]

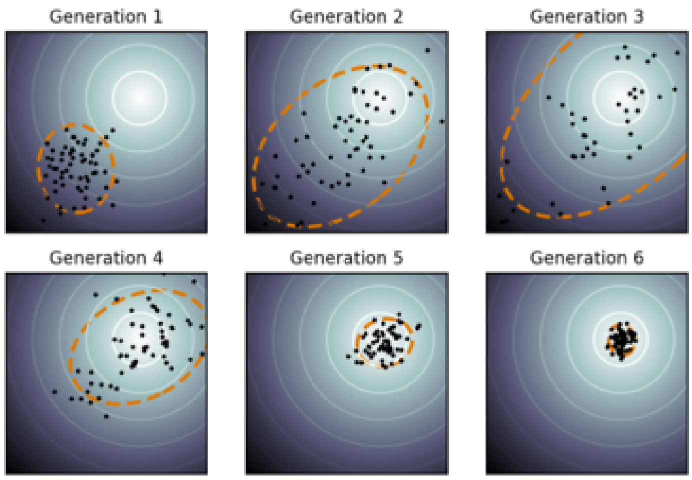
[3]Lindemann, D. L. (o. J.). Evolutionsstrategien [PDF]. Abgerufen 10. Dezember 2019, von http://ls11-www.cs.tu-dortmund.de/lehre/SoSe03/PG431/Ausarbeitungen/FolienEvolutionsstrategien.pdf

5. Method

The CMA-ES method is relatively simple when broken down. We are looking for the values with which the algorithm is successfully ended with a high or highest possible probability. Successfully this is defined differently from problem to problem and thus the values to be checked differ from problem to problem. Since we are not improving the algorithm itself, but rather the parameters with which the algorithm works, we can use CMA-ES here. The goal is to find the parameters with which the algorithm is ended as well as possible with the highest possible probability and finds an at least good solution. Better than the previous solution to a problem. The values we work with are therefore normal distributions which we combine, mutate and sort out. A combination of normal distributions becomes a joint probability [4] distribution (to be seen in Ill. 01).

In the example, we go from Ill. 01 to the 6th generation of the algorithm. However, in contrast to previous evolutionary strategies, one not only deals with the individual evolutionary steps but also examines the entire development of the algorithm. This is called the evolution path below.

The advantage of following evolutionary paths is that this type of ES maximizes the rate of progress. This is at the expense of the probability of selection. However, these algorithms create more optimized solutions. So you can not only compare individual steps but entire paths.



Wikipedia contributors. (2019). Ill. 01 [Illustration]

[4]Wikipedia contributors. (2019). Ill. 01 [Illustration]. Abgerufen von https://en.wikipedia.org/wiki/CMA-ES

If mutation steps selected in succession are correlated in parallel (scalar product greater than zero), the path of evolution is comparatively long. If the steps are correlated antiparallel (dot product less than zero), the evolutionary path is comparatively short. Parallel correlation means that successive steps run in the same direction. Anti-parallel correlation means that the steps cancel each other out. We assume that both correlations are inefficient. This is most evident when the correlation / counter-correlation between successive steps is perfect. These steps can be replaced exactly by the enlarged/reduced first step (Hansen and Ostermeier, no date).
So we start with the first generation. Here it can be seen that the algorithm was fed with data on which it should start.
Up to the third generation, the mutation strength is high. Because the points are spreading further and further. This is done to cover the widest possible range of possible optimizations. Then the mutation strength changes and thus the strategy after the mutations and subsequent selection happen. We are now trying to discuss the best options in the broad field. And then wants to focus on just these probabilities. So you specialize in the following generations. Here is the comparison of evolution paths clearly in the advantage as long-term movements can be noticed.
The goal is to be seen as in the last generation: Probabilities as high as possible to let the algorithm end with a good result.

6. Experiments

In the paper, the CMA-ES algorithm is tested for the Rastrigin function in different scales. For example, on page 32, we compare the CMA-Es algorithm with a MUT-ES algorithm. The function value is tested with the Rastrigin function with Scale 1000. The MUT-ES algorithm on the right side achieves a significantly worse result with a factor of 10,000.
But even at much smaller scales, the differences between the two algorithms are visible. On page 31 in the paper, you can see again the CMA-Es algorithm and the MUT-ES algorithm in direct comparison. If the two algorithms achieve approximately the same results for a simple Rastrigin function (again, the function value is tested), the differences in the optimization are already clearly recognizable with a Rastrigin function of 10. The MUT-IT performs 20 times worse than CMA-Es with the same underlying problem.

7. Discussion

Ultimately, the discussion is whether the strategy is to compare the evolutionary path and provides for increased optimization.
Based on the data collected in the experiments, a positive result can be advertised for these cases. The comparison of the evolutionary paths seems to be the perfect solution to further improve evolutionary algorithms. But as already explained in the Methods section, the comparison in the evolution path goes hand in hand with a smaller spread between the steps. this can lead to slower success, especially in the first generations.
It does not always seem to be the right way.

8. Conclusion

To take up the point of discussion again. The experiments show that the parameter strategy in the CMA-ES algorithm only pays off in complex orders of magnitude. In smaller problems, other algorithms such as the MUT-ES create the same or even better results because of the comparison in the entire evolution path as already mentioned at the expense of the spreads so the variety of the options goes. The paper itself states that problems have to be at least 10n (n is problem dimension) so that CMA-ES can really develop its potential and the distance to other algorithms becomes clear.

But especially with today's increasingly complex problems, this type of evolution strategy is an interesting and promising solution.