# Multi-objective Optimization with Unbounded Solution Sets

**Oswin Krause**
Dept. of Computer Science
University of Copenhagen
Copenhagen, Denmark
oswin.krause@di.ku.dk

**Tobias Glasmachers**
Institut für Neuroinformatik
Ruhr-Universität Bochum
Bochum, Germany
tobias.glasmachers
@ini.rub.de

**Christian Igel**
Dept. of Computer Science
University of Copenhagen
Copenhagen, Denmark
igel@di.ku.dk

## Abstract

Machine learning often requires the optimization of multiple, partially conflicting objectives. True multi-objective optimization (MOO) methods avoid the need to choose a weighting of the objectives *a priori* and provide insights about the trade-offs between the objectives. We extend a state-of-the-art derivative-free Monte Carlo method for MOO, the MO-CMA-ES, to operate on an unbounded set of (non-dominated) candidate solutions. The resulting algorithm, UP-MO-CMA-ES, performed well in two recent performance comparisons of MOO methods.

## 1 Introduction

Multi-objective optimization (MOO, also known as multi-criteria or vector optimization) addresses simultaneous optimization of several objectives. The goal is to find or to approximate the set of Pareto-optimal solutions. A solution is Pareto-optimal if it cannot be improved in one objective without getting worse in another one. There is an increasing interest in using MOO in machine learning [22, 23, 18], e.g., for parameter tuning and for understanding trade-offs. Conflicting objectives are ubiquitous, for example complexity vs. accuracy vs. computational complexity of big data processing [6], sensitivity vs. specificity (e.g., [30]), or performance on multiple tasks including multi-objective reinforcement learning (e.g., see [32] and references therein). While the attitude towards evolutionary optimization is sometimes marked by a touch of skepticism, multi-objective evolutionary algorithms (MOEAs) have become broadly accepted MOO methods. Model-based approaches such as ParEGO [24] are an attractive alternative for expensive problems, and so are evolutionary model-assisted algorithms [10]. Both types of methods typically fit global (regression) models to previous evaluations in order to predict unseen objective vectors. In contrast, the algorithm under study builds an ensemble of local models, and instead of inter- and extrapolating the objective functions, it learns distributions of promising new candidate solutions.

The *covariance matrix adaptation evolution strategy*, CMA-ES [15], is arguably one of the most competitive derivative-free algorithms for real-valued single-objective optimization [5, 11]. The algorithm has been successfully applied to many real-world problems, and its many variants have demonstrated superior performance in benchmarks and competitions. In machine learning, the CMA-ES is being applied for direct policy search in reinforcement learning and hyperparameter tuning in supervised learning (e.g., see [13, 12, 16, 17, 19], and references therein). There exists an extension of the CMA-ES to MOO, the multi-objective CMA-ES (MO-CMA-ES, [20, 33]). The original MO-CMA-ES uses an upper bound, given by the *offspring population size*, on the set of solutions it considers in each iteration. We recently developed a refined variant, termed *Unbounded Population MO-CMA-ES* (UP-MO-CMA-ES, [26]), which maintains an unbounded set of solutions, and submitted it to three benchmarking competitions. In the following, we present a generalization of the algorithm and summarize the benchmarking results.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

## 2 UP-MO-CMA-ES

The UP-MO-CMA-ES is the unbounded-population variant of MO-CMA-ES, which again is a multi-objective variant of the seminal CMA-ES algorithm. We start out by briefly sketching the working principles of these methods, and then describe UP-MO-CMA-ES.

**CMA-ES** The covariance matrix adaptation evolution strategy (CMA-ES, [15, 14]) is a Monte Carlo method for derivative-free single-objective optimization. In each iteration, it samples candidate solutions from a $d$-dimensional multivariate normal *search distribution*, and ranks them according to their objective function values. The mean and the covariance matrix of the search distribution are then adapted with various mechanisms based on the spatial distribution of the better-ranked points (cf. [15, 14]). Changing the mean corresponds to the actual optimization steps, while adapting the covariance matrix aims at maximizing the probability of sampling even better points. Adaptation of a global variance parameter enables evolution strategies to converge to the optimum at a linear rate [2, 21]. Adaptation of the full covariance turns CMA-ES into a variable metric methods, suitable for solving ill-conditioned problems. Key concepts of the update rule can be derived from principles from information geometry [1], resembling natural gradient learning. The CMA-ES comes with robust defaults for all of its hyperparameters, making it practically parameter-free. Efficient covariance matrix updates are discussed in [27, 25].

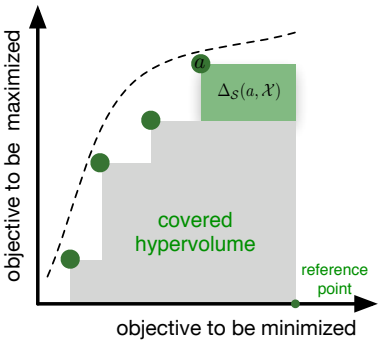


Figure 1: Contributed hypervolume $\Delta_S$ of a point $a$ to a set $\mathcal{X}$ of four solutions. The green and gray area are the hypervolume dominated by $\mathcal{X}$.

The search distribution in the CMA-ES does not model the objective function directly, but instead it models the algorithm's belief about promising new candidate solutions. However, it was found that on quadratic objective functions the (quadratic) log-density aligns with the objective, effectively performing model-guided optimization. Hence, despite performing direct search, CMA-ES is closely related to second order optimization methods like NEWUOA [29].

**MO-CMA-ES** The powerful covariance matrix adaptation principle has been adapted for solving MOO problems. The MO-CMA-ES algorithm [20, 33] maintains a set (population) of minimal CMA-ES instances (so-called 1+1 variants), covering different parts of the Pareto front (the objective function vectors of non-dominated solutions). Each CMA-ES instance maintains its own Gaussian distribution, forming a flexible mixture-of-Gaussians model of promising points along the arbitrarily shaped set of non-dominated solutions. The MO-CMA-ES differs from the CMA-ES mostly in its selection mechanism, which is based on maximizing the *dominated hypervolume* [3] (see Fig. 1) of the set of individuals in all CMA-ES instances [20, 33].

**Unbounded Population** The main differences of UP-MO-CMA-ES compared to MO-CMA-ES and most other MOEAs are the population model and the selection mechanism. The predominant paradigm has been to maintain a fixed number of candidate solutions. Newly generated solutions compete with existing ones based on their contributions to a set performance indicator, often the dominated hypervolume [3]. The fixed population size allows for sorting and ranking solutions by their contributions, a key concept for selection and adaptation of the search distribution. In contrast, UP-MO-CMA-ES maintains *all* non-dominated individuals in the population while discarding all dominated points. Hence the size of the population is dynamic and unbounded.

This design has consequences for the goal of optimization. At first the algorithm must approach the Pareto front. Once the front is reached, it does not follow the usual goal of finding the optimal distribution of a fixed number of $\mu$ points over the front, but aims at filling in the gaps. Hence, if successful, the hypervolume does not converge to the optimal $\mu$-distribution [3], but to the hypervolume covered by the actual (usually infinite) Pareto front. Of course, this requires unlimited memory, so in practice the algorithm must stop (for the latest) when it runs out of memory. This is not a serious limitation, we were able to run the algorithm in-memory on standard hardware on all BBOB-biobj 2016 problem instances for $10^6$ function evaluations.

An important component of the algorithm is the choice of the Gaussian distribution from which the next candidate point is sampled. We select the Gaussian distribution based on the hypervolume contribution of the corresponding point in the solution set, see Fig. 1. The contribution of the hypervolume provides a measure of how well the front is approximated locally. Thus, giving more weight to points with large contributions can lead to larger progress. The contributions of extreme solutions (the "ends" of the Pareto front, i.e., best points in one objective) critically depend on the somewhat arbitrary choice of a reference point (see Fig. 1). Therefore these points are selected differently, namely with a fixed probability. This helps the algorithm to spread the solution set over the whole Pareto front, irrespective of the size of the current set.

**Two Objectives** The UP-MO-CMA-ES for two objectives is described in [26]. The method exploits the one-dimensional and hence ordered structure of the Pareto front. Going beyond standard MO-CMA-ES, it averages (recombines) the covariance matrices of solutions that are neighbors in the current set of non-dominated solutions. Deciding whether a solution is dominated or not, updating the set accordingly, and computing all hypervolume contributions is particularly inexpensive in the case of two objectives. For $\mu$ non-dominated solutions this is achieved in $\mathcal{O}(\log(\mu))$ amortized time and $\mathcal{O}(\mu)$ memory [26].

**Extension to Three or More Objectives** For $m \geq 3$ objectives, there is no easy way to update the hypervolume contributions cheaply when a solution is inserted or removed. This is because the hypervolume contribution of an objective vector can be affected by arbitrarily many others. While even approximating the hypervolume is #P hard for growing $m$ [7], for $m = 3$ computing the hypervolume contributions (of one or all points) can be performed in $O(\mu \log(\mu))$ time, which still allows a large number of solutions in the front. We decided to select the Gaussian distribution corresponding to the point with maximum contribution as for $m > 3$ it is computationally cheaper to select (one of) the largest contributors with an approximation algorithm (e.g., [8, 9]) than to compute the contributions of all points with sufficient precision. In an ordered set each non-extreme solution has exactly two well-defined neighbors, to its left and to its right. There is no (parameter-free, canonical) extension of this concept to higher dimensions. Therefore we drop the recombination of covariance matrices, which anyway has a minor effect on performance.

## 3 Performance in Competitions

The UP-MO-CMA-ES was evaluated against several baselines and many strong competitors using two established benchmarking platforms. Detailed results tables and figures are found on the corresponding websites. Here we briefly describe the main results.

**Bi-objective Blackbox Optimization Benchmarking (BBOB)** The bi-objective version of the algorithm was run on the bi-objective version of the BBOB Benchmark [31].[1] The benchmark is based on a subset of functions of the single objective BBOB, where the bi-objective functions were generated by picking combinations of two functions. Therefore the benchmark contained function pairs as simple as two quadratic functions, but also pairs of multi-modal functions such as Rastrigin which are even challenging in the single-objective case.

We ran the UP-MO-CMA-ES on the full BBOB-biobj 2016 benchmark in dimensions $d \in \{2, 3, 5, 10, 20\}$ with a total budget of $10^6 d$ function evaluations. To handle multi-modality, the budget was distributed equally on 100 separate instances and the fronts merged after 1% of the total budget was spent.

In the benchmark our algorithm ranked second [26], after another MO-CMA-ES variant termed HMO-CMA-ES [28], which included a hybrid strategy targeted to solve highly multi-modal functions—and outperformed UP-MO-CMA-ES on such problem instances.

**Black Box Optimization Competition (BBComp)** In BBComp,[2] optimization problems are truly black boxes to participants. The evaluation budget is limited, not only for a "fi-

[1] A comparative review of the results is yet to be published, but see `http://coco.gforge.inria.fr/presentation-archive/2016-GECCO/11_Anne_bbob-2016-wrap-up.pdf`

[2] BBComp website with more details and all results: `http://bbcomp.ini.rub.de/`

nal run" of the algorithm, but overall. Hence, there is no opportunity for offline tuning to the specific problems at hand, so that the results are unbiased and unaffected by over-fitting of algorithm parameters to problems or problem classes. UP-MO-CMA-ES participated in the two and three objective tracks of the BBComp'2016 competition. On the two-objective track, UP-MO-CMA-ES ranked second out of 7 algorithms. It was outperformed by a hybrid method combining restarted local search and the SMS-EMOA algorithm [4]. On the three-objective track, UP-MO-CMA-ES ranked first out of 7 algorithms, beating the above mentioned heuristic with a large margin (see figure 2). We conclude that the unbounded population approach pays off in particular for higher dimensional Pareto fronts where filling in "holes" offers a large potential for gains.

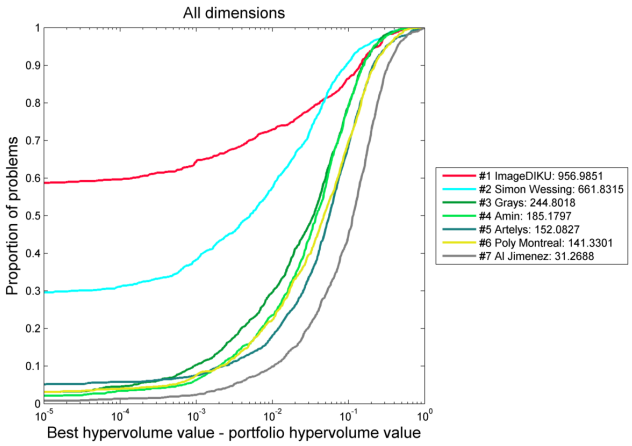


Figure 2: Aggregated performance of all seven algorithm in the three-objective track of BBComp, UP-MO-CMA-ES in red (higher and to the left is better).

For both tracks, a more detailed analysis based on problem dimension reveals sub-optimal parameter tuning for high dimensions. UP-MO-CMA-ES totally dominated the field in both tracks for low-dimensional problems with $d \leq 10$. However, it scored only second for $d \geq 32$. We suspect that for high-dimensional problems UP-MO-CMA-ES would profit from a more aggressive exploration strategy, using a larger share of the budget for locating interesting parts (local optima) of the search space.

## 4 Conclusions

Many machine learning problems are inherently multi-objective, and typically it is not clear how to weight these objectives *a priori*. Multi-objective optimization (MOO) algorithms allow to obtain a set of solutions approximating the Pareto optimal ones, those which cannot be improved in one objective without getting worse in another one. These solutions provide a basis for subsequent decision making and an understanding of the trade-off between partially conflicting objectives, for example training time, solution complexity, and predictive accuracy of learning machines. Evolutionary MOO algoritms (EMOAs) mark the state-of-the-art, in particular for non-expensive settings and studies without too limiting computation time constraints. Traditional EMOAs have been designed to operate on an *a priori* fixed number of active candidate solutions. Dropping this restriction removes one hyperparameter from the method, and more importantly, it allows to model the Pareto front more accurately. This modification was applied to one of the most elaborate EMOAs for continuous domains, the MO-CMA-ES. The optimization performance of the resulting UP-MO-CMA-ES was compared to the state-of-the-art on two benchmarking platforms, BBOB and BBComp. It demonstrated very promising performance. In its current form it apparently has two shortcomings: the BBOB results reveal that it can fail on highly multi-modal problems with millions of local optima. While we believe that the former class of optimization problems is not highly relevant, improved restart strategies and revised hyperparameter[3] settings may improve the performance of UP-MO-CMA-ES on such problems. The BBComp results show that the algorithm is not yet well-tuned for high-dimensional problems. We expect that this issues can be easily addressed by changing the way the dimensionality enters the algorithm's hyperparameters.

**Acknowledgements** Supported by the Innovation Fund Denmark through the *Danish Center for Big Data Analytics Driven Innovation* and the project *Personalized Breast Cancer Screening*.

[3]The hyperparameters in CMA-ES and MO-CMA-ES are typically not meant to be adjusted manually, but are set to robust default values that incorporate scaling with respect to $d$, $m$, etc.

## References


[1] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi. Theoretical foundation for CMA-ES from information geometry perspective. *Algorithmica*, 64(4):698–716, 2012.

[2] A. Auger. Convergence results for the $(1, \lambda)$-SA-ES using the theory of $\varphi$-irreducible Markov chains. *Theoretical Computer Science*, 334(1–3):35–69, 2005.

[3] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler. Theory of the hypervolume indicator: optimal $\mu$-distributions and the choice of the reference point. In *Foundations of Genetic Algorithms (FOGA X)*, pages 87–102. ACM Press, 2009.

[4] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[5] H.-G. Beyer. Evolution strategies. *Scholarpedia*, 2(8):1965, 2007.

[6] O. Bousquet and L. Bottou. The Tradeoffs of Large Scale Learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 161–168, 2008.

[7] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *International Symposium on Algorithms and Computation*, pages 436–447. Springer, 2008.

[8] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, 2012.

[9] K. Bringmann, T. Friedrich, C. Igel, and T. Voß. Speeding up many-objective optimization by Monte Carlo approximations. *Artificial Intelligence*, 204:22–29, 2013.

[10] A. Díaz-Manríquez, G. Toscano, J. H. Barron-Zambrano, and E. Tello-Leal. A review of surrogate assisted multiobjective evolutionary algorithms. *Computational Intelligence and Neuroscience*, 2016, 2016.

[11] A. E. Eiben and J. Smith. From evolutionary computation to the evolution of things. *Nature*, 521:476–482, 2015.

[12] T. Glasmachers and C. Igel. Uncertainty handling in model selection for support vector machines. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature (PPSN)*, pages 185–194. Springer, 2008.

[13] F. Gomez, J. Schmidhuber, and R. Miikkulainen. Accelerated neural evolution through cooperatively coevolved synapses. *Journal of Machine Learning Research*, 9:937–965, 2008.

[14] N. Hansen. The CMA evolution strategy: A tutorial. Technical report, Inria Saclay – Île-de-France, Université Paris-Sud, LRI, 2015.

[15] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[16] V. Heidrich-Meisner and C. Igel. Hoeffding and Bernstein races for selecting policies in evolutionary direct policy search. In *International Conference on Machine Learning (ICML 2009)*, pages 401–408, 2009.

[17] V. Heidrich-Meisner and C. Igel. Neuroevolution strategies for episodic reinforcement learning. *Journal of Algorithms*, 64(4):152–168, 2009.

[18] D. Horn, A. Demircioğlu, B. Bischl, T. Glasmachers, and C. Weihs. A Comparative Study on Large Scale Kernelized Support Vector Machines. *Advances in Data Analysis and Classification (ADAC)*, pages 1–17, 2016.

[19] C. Igel. Evolutionary kernel learning. In *Encyclopedia of Machine Learning and Data Mining*. Springer, 2016.

[20] C. Igel, N. Hansen, and S. Roth. Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28, 2007.

[21] J. Jägersküpper. Probabilistic runtime analysis of $(1+,\lambda)$, ES using isotropic mutations. In *Genetic and Evolutionary Computation Conference (GECCO)*, pages 461–468. ACM, 2006.

[22] Y. Jin, editor. *Multi-Objective Machine Learning*. Springer, 2006.

[23] Y. Jin and B. Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):397–415, 2008.

[24] J. Knowles. ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, 2006.

[25] O. Krause, D. R. Arbonès, and C. Igel. CMA-ES with optimal covariance update and storage complexity. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[26] O. Krause, T. Glasmachers, and C. Igel. Unbounded population MO-CMA-ES for the bi-objective BBOB test suite. In *Genetic and Evolutionary Computation Conference (GECCO 2016) Companion: BBOB-2016 Workshop*, pages 1177–1184. ACM, 2016.

[27] O. Krause and C. Igel. A more efficient rank-one covariance matrix update for evolution strategies. In *Foundations of Genetic Algorithms (FOGA XIII)*, pages 129–136. ACM, 2015.

[28] I. Loshchilov and T. Glasmachers. Anytime Bi-Objective Optimization with a Hybrid Multi-Objective CMA-ES (HMO-CMA-ES). In *Genetic and Evolutionary Computation Conference (GECCO 2016) Companion: BBOB-2016 Workshop*, pages 1169–1176. ACM, 2016.

[29] M. J. Powell. The NEWUOA software for unconstrained optimization without derivatives. In G. Di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 255–297. Springer, 2006.

[30] T. Suttorp and C. Igel. Multi-objective optimization of support vector machines. In Y. Jin, editor, *Multi-objective Machine Learning*, volume 16 of *Studies in Computational Intelligence*, pages 199–220. Springer, 2006.

[31] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The bi-objective black-box optimization benchmarking (bbob-biobj) test suite. *ArXiv e-prints*, arXiv:1604.00359, 2016.

[32] K. Van Moffaert and A. Nowé. Multi-objective reinforcement learning using sets of Pareto dominating policies. *Journal of Machine Learning Research*, 15:3663–3692, 2014.

[33] T. Voß, H. Hansen, and C. Igel. Improved step size adaptation for the MO-CMA-ES. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 487–494. ACM, 2010.