Improving Optimization With Gaussian Processes in the Covariance Matrix Adaptation Evolution Strategy

Jiří Tumpach[1], Jan Koza[2] and Martin Holeňa[3]

[1]Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic
[2]Czech Technical University, Faculty of Information Technology, Prague, Czech Republic
[3]Czech Academy of Sciences, Institute of Computer Science, Prague, Czech Republic

Abstract
This paper explores the use of Gaussian processes (GPs) in the covariance matrix adaptation evolution strategy (CMA-ES) for black-box optimization. GPs are powerful probabilistic models that capture complex relationships, making them suitable for modeling uncertain objective functions. Integrating GPs into the CMA-ES improves exploration and adaptation in the search space, enhancing convergence speed and solution quality. The paper describes a novel implementation framework allowing to use GPs as surrogate models for the CMA-ES. That framework findings encourage further research to advance the application of GPs in black-box optimization.

1. Introduction

Black-box optimization is an optimization of objective functions for which no analytical description is provided. It employs optimization methods that need as input only points in the search space paired with respective values of the objective function obtained in a non-analytical way, e.g. from sensors, in experiments or through numerical simulations. The most frequently used approaches are evolutionary optimization, such as evolution strategies, genetic algorithms, and differential evolution, or other metaheuristics, such as particle swarm optimization.

Because black-box optimization methods receive only information about values of the objective function, they typically need many such values. This is a problem in situations when evaluating the black-box objective function is time-consuming and/or expensive. That is frequently the case if it is evaluated empirically in experiments. For example, for the evolutionary optimization tasks described in the book [1], the evaluation of a comparatively small generation of a genetic algorithm can sometimes take more than a week and cost more than 10000 €. To tackle such problems, an approach called *surrogate modeling* has emerged more than 20 years ago. In particular in continuous optimization, surrogate modeling consists in evaluating the true, black-box objective function only in some points and evaluating a suitable regression model in all remaining points. Such a regression model is called *surrogate model* or *metamodel* of the objective function. It is trained on points where the true objective function has been evaluated and approximates it in the search space.

The earliest kinds of surrogate models in continuous black-box optimization were *low-order polynomials* and *artificial neural networks* (ANNs), specifically multilayer perceptron (MLP). The former have always remained a suitable choice in situations when enough evaluations of the true, black-box objective function are affordable for the approximation properties of polynomials to be in effect. On the other hand, surrogate modeling for substantially fewer evaluations of the true objective function has undergone further development during the last two decades. MLPs were soon replaced with another kind of ANNs, radial basis function networks (RBFs), which better fit the local peculiarities of an objective function landscape. Those networks, however, have since the late 2000s been superseded by other kinds of surrogate models, primarily *Gaussian processes* (GPs), but also ranking support vector machines (RSVMs) and random forests (RFs). GPs are currently the most successful kind of surrogate models for black-box optimization with a small evaluation budget of functions with complicated multimodal landscapes, mainly due to their ability to estimate the probability distribution of the true objective function in a given point.

2. Surrogate Modeling in Black-Box Optimization

Surrogate modeling for black-box optimization relies on the combination and interaction of three components: a *regression model* serving as a surrogate of the true, black-box objective function, a *black-box optimization method* seeking the optimum of that objective function, and a strategy when to evaluate the true objective function and when its surrogate model. In the context of evolutionary black-box optimization, that strategy is usually called *evolution control* [2, 3, 4, 5, 6].

The regression models that are the most suitable kind

ITAT'23: Information Technologies – Applications and Theory, 11th International Workshop on Computational Intelligence and Data Mining

of surrogate models if sufficiently many evaluations of the true, black-box objective function are affordable, are low-order polynomials, typically quadratic functions [7, 8, 9, 10, 11]. The sufficient number of evaluations depends, according to these cited research works, on the black-box function and on the dimension. For substantially fewer evaluations, the most traditional kind of surrogate models were MLPs [5, 12], soon replaced with RBFs [13, 14, 11, 15, 10], and since the late 2000s with Gaussian processes (GPs) a.k.a. kriging [2, 4, 16, 17, 18, 19, 20, 21, 22]. Occasionally, RBFs were used as local models in combination with GP-based global models [23]. Other kinds of surrogate models employed during the last decade include decision trees [24], random forests [25, 26, 24] and ranking support vector machines [27, 28]. The last one has an exceptional property of *invariance with respect to order-preserving transformations* of the objective function. This is important in situations when the black-box optimization algorithm possesses such invariance, a frequently encountered property of evolutionary algorithms. On the other hand, the surrogate modeling methods proposed in [4] and [22] use GPs to perform preselection based on a partial ordering that is also invariant with respect to order-preserving transformations. More importantly, the adaptive function value warping approach recently proposed in [29] aims to provide such invariance to any surrogate model.

As to the black-box optimization methods, surrogate models are most often combined with evolutionary optimizers. Their combinations with the most important among them, the state-of-the-art black-box optimization algorithm CMA-ES will be surveyed in some detail below, in Subsection 2.1. GPs were combined also with other evolutionary optimization methods [18, 30], and GPs, polynomials and RBFs were combined with particle swarm optimization [11] and with memetic optimization [14]. Moreover, GPs are used in black-box optimization in two different ways. In connection with evolutionary and similar black-box optimization methods, they serve as a regression model evaluated instead of the true objective function. In addition, they also play a key role in *Bayesian optimization.* That kind of optimization relies on GP-estimates of probability distributions of values of the true objective function. Those probability distributions enable several ways of searching for optima of that objective function, each of them governed by a specific *acquisition function* [31, 32, 33]. The surrogate-assisted black-box optimization methods constructing several surrogate models simultaneously either aggregate them to a team [14, 11] or complement the evolution control by a classifier selecting the most appropriate among those models. Important examples of classifiers used in this context are ANNs [34, 35, 36] and classification trees [37, 20]. Their learning can be viewed as *metalearning* because it is based on *metafeatures*, i.e. properties empirically characterizing the objective function landscape and the black-box optimization method [35, 24, 38, 10]. Apart from classification according to the appropriateness of the surrogate model for the considered data, metalearning can also be used for regression of model error on the combination of values of metafeatures [39].

Finally, evolution control has been since the first surrogate-assisted black-box optimization methods performed basically in two ways, *generation-based* and *individual-based.* In the generation based, all points are in some generations evaluated with the true objective function and in the remaining generations with the model. On the other hand, in every generation of the individual-based evolution control, based on the evaluation of all points with the model, a preselection of points to be evaluated with the true objective function is performed [5]. In most of the surrogate-assisted methods, however, the evolution control is specifically tailored to the respective method.

2.1. Surrogate Modeling in Connection With the CMA-ES

Not only the two most important kinds of surrogate models, i.e. low-order polynomials [7, 8, 9] and GPs [13, 4, 17, 21, 22], but also the less common RBFs, RFs and RSVMs [25, 27, 26, 15] are most often combined with the *Covariance matrix adaptation evolution strategy* (CMA-ES). That is not surprising because CMA-ES has already in the 2000s become a state-of-the-art approach to single-objective unconstrained continuous black-box optimization [40, 41]. Occasionally, also Bayesian optimization is combined with CMA-ES. For example in [42], optimization switches from the most traditional Bayesian optimization method, EGO (Efficient Global Optimization) [32], to CMA-ES. Finally, CMA-ES has also been combined with a team of surrogate models and the choice of the most appropriate among them based on landscape analysis [37, 20].

As to the evolution control of surrogate-assisted variants of CMA-ES, the authors of the present paper have been involved into an investigation of the evolution control of two important polynomial-assisted CMA-ES variants lmm-CMA [7, 9] and lq-CMA-ES [8] and of two variants of the GP-assisted variant DTS-CMA-ES [2, 19]. Noteworthy, that investigation included mutually replacing the evolution control of each variant with the evolution control of the others. According to its findings, the success of those important surrogate-assisted CMA-ES variants is definitely not limited to using the respective specific tailored evolution control [6].

3. New Framework for a Surrogate-Assisted CMA-ES

The most widely used implementation of the CMA-ES algorithm is the official code written by the author of the algorithm Nikolaus Hansen and his team [43]. It is available in multiple programming languages, including C, C++, Matlab, R, Python, and others. It is being actively developed, and it contains various versions and extensions of the algorithm and extensive parameterization options. While the C and C++ versions are the most performant for solving real problems in practice, the most suitable for experimentation with the algorithm itself is nowadays the Python version. However, the Python CMA-ES version is still based on the original Matlab legacy code rewritten into Python. It contains very long function definitions with multiple nested if statements for different algorithm variants and parameter handling, which makes it highly inconvenient to experiment with modifications of the core parts of the algorithm.

Therefore we decided to base our code on a different implementation by Jacob de Nobel and his colleagues called Modular CMA-ES [44], which is written in a modern modular object-oriented way, allowing to create different variants of the CMA-ES algorithm easily.

3.1. Modular CMA-ES

The starting point of our implementation is the library Modular CMA-ES. Each optimization technique is encapsulated within a modular component, providing independence and flexibility in selecting and combining different modules. This modularity enables users to construct tailored optimization strategies by combining multiple modules, thereby expanding the exploration space and enhancing the search capabilities of the CMA-ES algorithm. By integrating previously distant optimization techniques, the library enables combinatorial exploration of different strategies within the CMA-ES framework. Users can effortlessly combine modules representing various optimization methods such as population sampling techniques, surrogate modeling, elitism, step size adaptation, restart strategies, and constraint handling mechanisms. This combinatorial exploration empowers researchers to exploit the strengths of different techniques, leading to more effective and efficient optimization processes. The Modular CMA-ES library prioritizes ease of use and customization. Moreover, the modular architecture allows for the activation and deactivation of modules during runtime, facilitating dynamic exploration and adaptation during the optimization process.

A general scheme of an evolution strategy can be expressed in the following steps:

1. Generate a new population
2. Evaluate individuals
3. Select parents
4. Reproduce
5. Repeat

For the CMA-ES algorithm in particular, the steps are these:

1. Sample λ points x_i
2. Evaluate the objective function $f(x_i)$
3. Select μ lowest $f(x_i)$
4. Update the population mean and covariance matrix
5. Repeat until optimum reached

These steps correspond to the methods implemented in the main class of the framework *ModularCMAES* as shown in the diagram in Figure 1. It also depicts the so-called ask-and-tell interface provided by the framework as well.

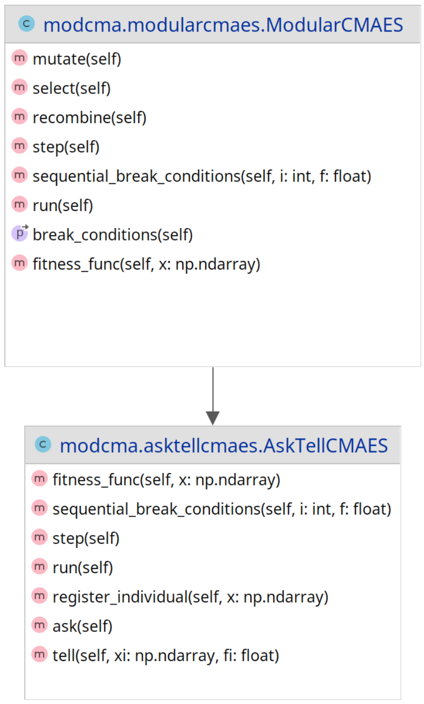

Figure 1: UML of the main classes in the Modular CMA-ES framework, which serves as the interface between the problem (objective function f) and the solver (CMA-ES)

However, this library does not provide support for surrogate models on its own. That is why we have been developing the framework described in this paper.

3.2. Incorporating Gaussian Processes

We added to the Modular CMA-ES package popular covariance functions such as Matérn, RBF, periodic, and many others [45]. In addition to these individual kernels, the package also provides the flexibility to explore additive and multiplicative combinations of them, cf. Subsection 3.3. This allows users to create more complex and customized GP-based surrogate models by combining multiple kernels together. Furthermore, the framework offers a search within these kernels. A list of Gaussian process covariance functions that are available in the framework follows.

Included covariance functions [45]

- Polynomial Kernels
- Parabolic
- RBF
- Exponential curve
- Periodic kernel
- Matérn$_{\frac{1}{2}}$, Matérn$_{\frac{3}{2}}$ and Matérn$_{\frac{5}{2}}$

Included covariance function modifications

- Learnable scaling of features
- Exponential mapping

3.3. A Systematic Approach to Combining Incorporated Covariance Functions

The works [46] and in more detail [47] present a systematic approach to automating the construction of GP covariance functions. Compositional kernels enable flexible and automatic discovery of the appropriate structure and complexity of a model by allowing the composition of multiple simpler kernels. By combining these kernels, the model can capture a wide range of patterns and structures, adapting to the complexity of the underlying data. Our framework evaluates the performance of each kernel through cross-validated regression, ensuring its effectiveness in capturing the underlying data patterns. Additionally, a complexity-based penalization approach is employed to assess the complexity of each kernel. By incorporating these evaluation methods, the framework enables the automatic selection of the most suitable kernels for optimizing complex problems.

3.4. Included Evolution Control

Evolution control in surrogate CMA-ES involves the management of the surrogate model and the decision-making process of how to update it. The key idea is to balance the exploration of the search space and the exploitation of promising regions guided by the surrogate model's predictions. The evolution control in surrogate CMA-ES plays a crucial role in leveraging the surrogate model to guide the search. We will briefly outline two different evolution controls we implemented in the framework.

Doubly Trained S-CMA-ES

The DTS-CMA-ES published in [2, 19] is a successor to the S-CMA-ES algorithm, which it extends with a second round of surrogate model training. The algorithm involves sampling a new population, training a surrogate model on original-evaluated points, selecting points based on the model's prediction, evaluating those points, retraining the model, and predicting fitness for non-original evaluated points. The key features include sampling from the CMA-ES distribution, utilizing Gaussian process uncertainty estimation for point selection, using recent points for fitness prediction, and maintaining a training set near the CMA-ES distribution mean.

Each generation of this EC can be summarized in the following steps:

1. sample a new population of size λ (standard CMA-ES offspring),
2. train the *first* surrogate model on the original-evaluated points from the archive A,
3. select $\lceil \alpha\lambda \rceil$ point(s) wrt. a criterion C, which is based on the *first* model's prediction,
4. evaluate these point(s) with the original fitness,
5. retrain the surrogate model also using these new point(s), and
6. predict the fitness of the non-original evaluated points with this *second* model.

Kendall-τ Rank Test Strategy From lq-CMA-ES

In this evolution strategy developed for the surrogate-assisted CMA-ES variant LQ CMA-ES [8], which is based on quadratic polynomials, a queue is utilized to store all evaluated solutions for model building. During each iteration, a limited number of the best solutions based on the model's performance are chosen from the population. These selected solutions are then evaluated using the true objective function f, sorted, and added to the end of the queue (with the best solution being enqueued last). To maintain the queue's size, the oldest elements are dropped when the maximum capacity is reached. This process continues until the Kendall-τ rank correlation coefficient between the rankings of function f and the model's rankings exceeds a threshold of 0.85, or until the entire population has been evaluated. At the end of the process, the population is ranked based on surrogate fitness unless all population members have been evaluated using function f, in which case the rankings based on function f are used. Through using the correlation coefficient, this approach avoids a direct comparison of the model and true objective function.

3.5. IOHprofiler Integration

The use of Modular CMA-ES in conjunction with IOHprofiler [48, 49] offers a powerful approach for analyzing and comparing iterative optimization heuristics. IOHprofiler, a versatile tool for evaluating algorithm performance, provides statistical assessments by analyzing the distribution of fixed-target running time and fixed-budget function values. By integrating modular CMA-ES with IOHprofiler, researchers can gain insights into the algorithm's behavior, assess its adaptability, and compare its performance against other optimization heuristics. The combination allows for tracking the evolution of algorithm parameters, facilitating the analysis, comparison, and design of self-adaptive algorithms. With IOHprofiler's experimental and post-processing capabilities, researchers can generate and evaluate running time data for benchmark problems, adjust the precision and range of displayed data, and make informed decisions based on the statistical evaluations produced.

4. Conclusion

This paper presented a new framework for support of the state-of-the-art black-box optimization method CMA-ES through GP-based modeling. It is a work-in-progress paper: not all intended functionality described in Section 3 has already been implemented and even some of the implemented is not yet working properly. However, we hope that the situation will be much better at the time of the workshop. Still, we are not aware of any other system that provides such a comprehensive functionality for combining CMA-ES with Gaussian processes.

We have concentrated on Gaussian processes because we consider them to be the most suitable kind of surrogate models for difficult multimodal black-box functions if only a small number of evaluations of the true objective function is available. In the future, however, we intend to extend the developed framework also to other kinds of surrogate models. Most importantly, to low-order polynomials, which are a surrogate-modeling continuation of traditional response surface models [50], and which have always been the most successful kind of surrogate models if a large number of evaluations of the true objective function is available or if that function is easy to fit. In addition, we intend to include also some other of the models recalled above in Section 2, as well as several models that have not yet been employed for surrogate modeling, but we believe that they are worth to be investigated to this end. For various time horizons, we think altogether of the following models:

- *Deep Gaussian processes*, in which an ANN architecture connects individual GPs, similarly to connecting individual recurrence cells in a long short term memory [51, 52].
- MLPs in the *neural tangent kernel* parametrization [53, 54, 55], which at a sufficient width have an ability to mimic GP sampling and to replace traditional acquisition functions in Bayesian optimization. Such behaviour of this kind of ANNs is, according to [53] and [55], a consequence of their asymptotic properties if the number of hidden neurons increases to infinity [56, 54, 57].
- *Variational autoencoders*, allowing to perform optimization on a latent space of a substantially lower dimension. Such use of a low dimensional latent space has already been investigated in the case of Bayesian optimization [58, 59].
- The *generative adversarial networks (GANs)* paradigm has been recently shown to be applicable to black-box optimization. More precisely, a generator has to propose samples compatible with the distribution of low values or directly with the distribution of the optimum of the considered black-box function, whereas one or more discriminators have to classify samples according to whether they are governed by that distribution [60, 61].

Acknowledgments

This work was supported by the Czech Technical University grant SGS23/205/OHK3/3T/18 and by the SVV project number 260 575 of the Charles University. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA LM2018140).

References

[1] M. Baerns, M. Holeňa, Combinatorial Development of Solid Catalytic Materials. Design of High-Throughput Experiments, Data Analysis, Data Mining, Imperial College Press / World Scientific, London, 2009.

[2] L. Bajer, Z. Pitra, J. Repický, M. Holeňa, Gaussian process surrogate models for the CMA evolution strategy, Evolutionary Computation 27 (2019) 665–697.

[3] D. Büche, N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with Gaussian process fitness function models, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 35 (2005) 183–194.

[4] C. Huang, B. Radi, A. El Hami, H. Bai, CMA evolution strategy assisted by kriging model and ap-

proximate ranking, Applied Intelligence 48 (2018) 4288–4204.
[5] Y. Jin, M. Olhofer, B. Sendhoff, A framework for evolutionary optimization with approximate fitness functions, IEEE Transactions on Evolutionary Computation 6 (2002) 481–494.
[6] Z. Pitra, M. Hanuš, J. Koza, J. Tumpach, M. Holeňa, Interaction between model and its evolution control in surrogate-assisted CMA evolution strategy, in: GECCO, 2021, p. 358 (paper no.).
[7] A. Auger, D. Brockhoff, N. Hansen, Benchmarking the local metamodel cma-es on the noiseless BBOB'2013 test bed, in: GECCO, 2013, pp. 1225–1232.
[8] N. Hansen, A global surrogate assisted CMA-ES, in: GECCO, 2019, pp. 664–672.
[9] S. Kern, N. Hansen, P. Koumoutsakos, Local metamodels for optimization using evolution strategies, in: PPSN, 2006, pp. 939–948.
[10] H. Yu, C. Sun, Y. Tan, J. Zeng, Y. Jin, An adaptive model selection strategy for surrogate- assisted particle swarm optimization algorithm, in: IEEE SCI, 2016, pp. 1–8.
[11] H. Wang, Y. Jin, J. Doherty, Committee-based active learning for surrogate-assisted particle swarm optimization of expensive problems, IEEE Transactions on Cybernetics 47 (2017) 2664–2677.
[12] M. Papadrakakis, N. Lagaros, Y. Tsompanakis, Structural optimization using evolution strategies and neural networks, Computer Methods in Applied Mechanics and Engineering 156 (1998) 309–333.
[13] L. Bajer, M. Holeňa, Surrogate model for continuous and discrete genetic optimization based on RBF networks, in: Intelligent Data Engineering and Automated Learning, Springer, 2010, pp. 251–258.
[14] D. Lim, Y. Ong, Y. Jin, S. B., A study on metamodeling techniques, ensembles, and multi-surrogates in evolutionary computation, in: GECCO, 2007, pp. 1288–1295.
[15] H. Ulmer, F. Streichert, A. Zell, Model-assisted steady state evolution strategies, in: GECCO, Springer, 2003, pp. 610–621.
[16] O. Krause, Recombination weight based selection in the DTS-CMA-ES, in: PPSN, 2022, pp. 295–308.
[17] Z. Li, T. Gao, B. Wang, Elite-driven surrogate assisted CMA-ES algorithm by improved lower confidence bound method, in: Engineering with Computers, Springer, 2022, pp. 10.1007/s00366–022–01642–5 (doi).
[18] L. Na, Q. Feng, Z. Liang, W. Zhong, Gaussian process assisted coevolutionary estimation of distribution algorithm for computationally expensive problems, Journal of Central South University of Technology 19 (2012) 443–452.
[19] Z. Pitra, L. Bajer, M. Holeňa, Doubly trained evolution control for the surrogate cma-es, in: PPSN, 2016, pp. 59–68.
[20] Z. Pitra, J. Repický, M. Holeňa, Landscape analysis of Gaussian process surrogates for the covariance matrix adaptation evolution strategy, in: GECCO, ACM, 2019, pp. 691–699.
[21] L. Toal, D. Arnold, Simple surrogate model assisted optimization with covariance matrix adaptation, in: PPSN, 2020, pp. 184–197.
[22] V. Volz, G. Rudolph, B. Naujoks, Investigating uncertainty propagation in surrogate-assisted evolutionary algorithms, in: GECCO, 2017, pp. 881–888.
[23] Z. Zhou, Y. Ong, P. Nair, A. Keane, K. Lum, Combining global and local surrogate models to accellerate evolutionary optimization, IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews 37 (2007) 66–76.
[24] B. Saini, M. López-Ibañez, K. Miettinen, Automatic surrogate modelling technique selection based on features of optimization problems, in: GECCO, 2019, pp. 1765–1772.
[25] N. Belkhir, J. Dréo, P. Savéant, M. Schoenauer, Per instance algorithm configuration of CMA-ES with limited budget, in: GECCO, 2017, pp. 681–688.
[26] Z. Pitra, J. Repický, M. Holeňa, Boosted regression forest for the doubly trained surrogate covariance matrix adaptation evolution strategy, in: ITAT 2018, 2018, pp. 72–79.
[27] I. Loshchilov, M. Schoenauer, M. Sebag, Comparison-based optimizers need comparison-based surrogates, in: PPSN, 2010, pp. 364–373.
[28] T. Runarsson, Ordinal regression in evolutionary computation, in: PPSN, 2006, pp. 1048–1057.
[29] A. Abbasnejad, D. V. Arnold, Adaptive function value warping for surrogate model assisted evolutionary optimization, in: Parallel Problem Solving from Nature – PPSN XVII: 17th International Conference, PPSN 2022, Dortmund, Germany, 2022, pp. 76–89.
[30] M. Wu, A. Karkar, B. Liu, A. Yakovlev, G. Gielen, V. Grout, Network on chip optimization based on surrogate model assisted evolutionary algorithms, in: IEEE CEC, 2014, pp. 3266–3271.
[31] Y. Diouane, V. Picheny, R. Le Riche, A. Di Perrotolo, TREGO: a trust-region framework for efficient global optimization, Journal of Global Optimization 85 (2022) 10.1007/s10898–022–01245–w (doi).
[32] D. Jones, M. Schonlau, W. Welch, Efficient global optimization of expensive black-box functions, Journal of Global Optimization 13 (1998) 455–492.
[33] J. Knowles, ParEGO: a hybrid algorithm with online landscape approximation for expensive multiobjective optimization problems, IEEE Transactions on Evolutionary Computation 10 (2006) 50–66.
[34] Y. He, Y. Yuen, Black box algorithm selection by

convolutional neural network, in: LOD, 2020, pp. 264–280.
[35] M. Pikalov, V. Mironovich, Automated parameter choice with exploratory landscape analysis and machine learning, in: GECCO, 2021, pp. 1982–1985.
[36] R. Prager, M. Seiler, H. Trautman, P. Kerschke, Towards feature-free automated algorithm selection for single-objective continuous black box optimization, in: IEEE SCI, 2021, pp. 1–8.
[37] Z. Pitra, L. Bajer, M. Holeňa, Knowledge-based selection of gaussian process surrogates, in: ECML Workshop IAL, 2019, pp. 48–63.
[38] R. Seiler, M.V.and Prager, P. Kerschke, H. Trautmann, A collection of deep learning-based feature-free approaches for characterizing single-objective continuous fitness landscapes, in: GECCO, 2022, pp. 657–665.
[39] A. Jankovic, G. Popovski, T. Eftimov, C. Doerr, The impact of hyper-parameter tuning for landscape-aware performance regression and algorithm selection, in: GECCO, 2021, pp. 687–696.
[40] N. Hansen, A. Ostermaier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 9 (2001) 159–195.
[41] N. Hansen, The CMA evolution strategy: A comparing review, in: Towards a New Evolutionary Computation, Springer, 2006, pp. 75–102.
[42] H. Mohammadi, R. Riche, E. Touboul, Making EGO and CMA-ES complementary for global optimization, in: Learning and Intelligent Optimization, Springer, 2015, pp. 287–292.
[43] N. Hansen, CMA-ES source code, 2016. https://cma-es.github.io.
[44] J. de Nobel, D. Vermetten, The source code of the modular python version of CMA-ES, 2020. https://github.com/IOHprofiler/ModularCMAES.
[45] E. Rasmussen, C. Williams, Gaussian Processes for Machine Learning, MIT Press, Cambridge, 2006.
[46] D. Duvenaud, J. Lloyd, R. Grosse, J. Tenebaum, G. Zoubin, Structure discovery in nonparametric regression through compositional kernel search, in: 30th International Conference on Machine Learning, 2013, pp. 1166–1174.
[47] D. Duvenaud, Automatic Model Construction with Gaussian Processes, Ph.D. thesis, University of Cambridge, 2014.
[48] C. Doerr, H. Wang, F. Ye, S. van Rijn, T. Bäck, IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics, 2018. Arxiv 1810.05281.
[49] C. Doerr, F. Ye, N. Horesh, H. Wnag, O. Shir, et al., Benchmarking discrete optimization heuristics with IOHprofiler, Applied Soft Computing Journal 88 (2020) 106027 (paper no.).
[50] R. Myers, D. Montgomery, C. Anderson-Cook, Response Surface Methodology: Proces and Product Optimization Using Designed Experiments, John Wiley and Sons, Hoboken, 2009.
[51] T. Bui, D. Hernandez-Lobato, J. Hernandez-Lobato, Y. Li, R. Turner, Deep Gaussian processes for regression using approximate expectation propagation, in: ICML, 2016, pp. 1472–1481.
[52] G. Hernández-Muñoz, C. Villacampa-Calvo, D. Hernández Lobato, Deep Gaussian processes using expectation propagation and Monte Carlo methods, in: ECML PKDD 2020, 2021, pp. 479–494.
[53] B. He, B. Lakshminarayanan, Y. Teh, Bayesian deep ensembles via the neural tangent kernel, in: NIPS, 2020, pp. 1–13.
[54] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks, in: NIPS, 2018, pp. 1–10.
[55] B. Paria, B. Pòczos, K. Ravikumar, S. J., A. Suggala, et al., Be greedy –- a simple algorithm for black-box optimization using neural networks, in: ICML Workshop on Adaptive Experimental Design and Active Learning in the Real World, 2022, pp. 1–27.
[56] S. Arora, S. Du, W. Hu, Z. Li, R. Salakhutdinov, et al., On exact computation with an infinitely wide neural net., in: NIPS, 2019, pp. 1–10.
[57] J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, et al., Wide neural networks of any depth evolve as linear models under gradient descent, in: NIPS, 2019, pp. 1–10.
[58] S. Kim, P. Lu, C. Lob, J. Smith, J. Snoek, et al., Deep learning for bayesian optimization of scientific problems with high-dimensional structure, Transactions on Machine Learning Research 1 (2023) open-review tPMQ6Je2rB.
[59] A. Tripp, E. Daxberger, J. Hernández-Lobato, Sample-efficient optimization in the latent space of deep generative models viaweighted retraining, in: NIPS, 2020, pp. 1–14.
[60] M. Gillhofer, H. Ramsauer, J. Brandstetter, B. Schäfl, S. Hochreiter, A GAN based solver of black-box inverse problems, in: NIPS, 2019, pp. 1–5.
[61] M. Lu, S. Ning, S. Liu, F. Sun, B. Zhang, et al., OPT-GAN: A broad-spectrum global optimizer for black-box problems by learning distribution, 2022. Arxiv 2102.03888v5.