

Towards an FCA-based Recommender System for Black-Box Optimization

Josefine Asmus^{1,3}, Daniel Borchmann², Ivo F. Sbalzarini^{1,3}, and
Dirk Walther^{2,3}

¹ MOSAIC Group, Center for Systems Biology Dresden (CSBD), Max Planck
Institute of Molecular Cell Biology and Genetics, Dresden, Germany.

² TU Dresden, Theoretical Computer Science, Dresden, Germany.

³ Center for Advancing Electronics Dresden (cfaED), Dresden, Germany.

{asmus,ivos}@mpi-cbg.de

{Daniel.Borchmann,Dirk.Walther}@tu-dresden.de

Abstract. Black-box optimization problems are of practical importance throughout science and engineering. Hundreds of algorithms and heuristics have been developed to solve them. However, none of them outperforms any other on all problems. The success of a particular heuristic is always relative to a class of problems. So far, these problem classes are elusive and it is not known what algorithm to use on a given problem. Here we describe the use of Formal Concept Analysis (FCA) to extract implications about problem classes and algorithm performance from databases of empirical benchmarks. We explain the idea in a small example and show that FCA produces meaningful implications. We further outline the use of attribute exploration to identify problem features that predict algorithm performance.

1 Introduction

Optimization problems are ubiquitous in science and engineering, including optimizing the parameters of a model [11], finding an optimal statistical estimator, or finding the best operating conditions for an electronic circuit or a biochemical network. An optimization problem is defined by a parameter space spanned by the variables to be optimized, and an objective function defined over that space. The goal is to find points in the parameter space where the objective function has an extremum. In a black-box problem, the function is not known, but it can be evaluated point-wise. The objective function is hence given as an oracle, which, given a point in parameter space as an input, returns the objective function value at that point. Most practical applications in science and engineering are black-box problems, where the objective function may comprise running a numerical computer simulation or performing a laboratory measurement in order to, e.g., determine how well an electronic circuit works.

Partially funded by the German Research Foundation (DFG) via the Cluster of Excellence ‘Center for Advancing Electronics Dresden (cfaED)’. Supported by DFG Research Training Group 1763 (QuantLA).

Owing to their practical importance, hundreds of algorithms and heuristics have been developed to (approximately) solve black-box optimization problems. The “No Free Lunch Theorem” [19] states that no algorithm performs better (i.e., finds the extremum in less iterations or finds a better extremum) than any other algorithm on all problems. Algorithm performance is hence relative to a problem or a class of problems. It is, however, unknown what these problem classes are, and how one should choose a “good” algorithm given a certain problem instance.

We aim at developing a recommender system for proposing efficient algorithms for a given black-box optimization problem. Formal Concept Analysis (FCA) [6] has previously been shown useful in recommender systems for rating systems and advertisement [4, 8]. Here we present the idea of using attribute exploration in FCA to find discriminative attributes of empirical benchmark data of different algorithms tested on different problems. We show that FCA produces meaningful implications about problem features and algorithms that imply good performance. Some of the implications found confirm hypotheses that are commonly known in the optimization community, others are novel. We also outline the use of attribute exploration to discover sets of problem features that are particularly predictive for algorithm performance. This is work in progress and we are not presenting a final solution. We do, however, believe that FCA will provide a powerful tool on the way towards a generic recommender system and problem classification for black-box optimization.

2 Black-box Optimization Problems

We consider real-valued scalar black-box problems, modeled as an oracle function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ mapping a d -tuple \mathbf{x} of real numbers (problem parameters) to a scalar real number $f(\mathbf{x})$ (objective function value). The function f is not assumed to be explicitly known. A *black-box optimization problem* entails finding extremal points of f using only point-wise function evaluations. The iterative sequence of evaluations used to progress toward an extremum is called the *search strategy*.

The Black-Box Optimization Benchmark (BBOB) test suit is a standard collection of test functions used to empirically benchmark different optimization heuristics [7]. The suit contains 24 benchmark functions with known ground truth for parameter spaces of different dimensions. Figure 1 plots four example functions in two dimensions.

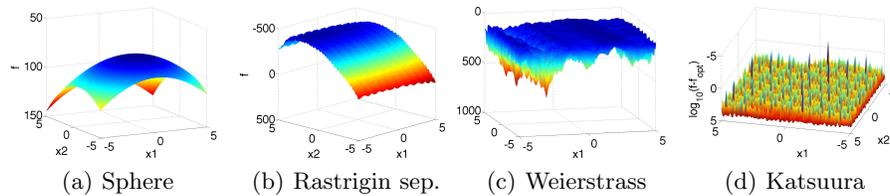


Fig. 1. Four black-box optimization benchmark (BBOB) functions in two dimensions.

A *problem instance* is a triple (f, d, ϵ) , where f is an instance of a black-box function, $d > 0$ is the parameter space dimension, and $\epsilon \in \mathbb{R}_{\geq 0}$ a tolerance.

Solving a problem instance (f, d, ϵ) means finding a d -tuple $\mathbf{x} \in \mathbb{R}^d$ such that $|f_{\min} - f(\mathbf{x})| < \epsilon$, where f_{\min} is the global minimum of f . Note that the ground-truth global minimum f_{\min} is known for the benchmark functions, but not in a practical problem instance.

3 An FCA-based Recommender System

We propose a system for recommending suitable algorithms for a given black-box optimization problem based on FCA. Similar recommender systems, albeit not based on FCA, have been suggested [3, 9, 15], and FCA has been applied to collaborative recommender systems for high-dimensional ratings [4] and in contextual advertising [8]. To the best of our knowledge, the use of FCA in a recommender system for black-box optimization has not been attempted so far.

3.1 Performance Measures

A run of an algorithm solving a problem instance is called *successful*. The performance of an algorithm a on a problem instance p is measured by the *expected running time* (ERT) [1, 3]:

$$\text{ERT}(a, p) = N^+(a, p) + \frac{1 - \pi^+(a, p)}{\pi^+(a, p)} N^-(a, p),$$

where $N^+(a, p)$ and $N^-(a, p)$ are the average numbers of function evaluations for successful and unsuccessful runs of a on p , respectively, and $\pi^+(a, p)$ is the ratio of the number of successful runs over the number of all runs of a on p .

The problem instances (f_j, d, ϵ) , where $j \in \{1, \dots, 24\}$, $d \in \{2, 3, 5, 10, 20, 40\}$, and $\epsilon \in E = \{10^3, 10^1, 10^{-1}, 10^{-3}, 10^{-5}, 10^{-8}\}$, have been used to test over 112 algorithms; the dataset is available online.¹ Note that the algorithms treat the benchmark functions as black boxes, i.e., they only evaluate the functions, whereas the actual function remains hidden to the algorithm. The known global minimum is only used to evaluate the performance of the algorithms.

Often, algorithms cannot solve problem instances for any tolerance value within a certain number of function evaluations; see Table 1. To obtain a performance measure also in these cases, we introduce a penalty of 10^6 for every tolerance level that the algorithm could not solve. This *penalized ERT* (pERT) of an algorithm a on an instance f of a benchmark function in d dimensions for a set E of tolerances is here defined as:

$$\text{pERT}(a, f, d, E) = \frac{\text{ERT}(a, (f, d, \epsilon_{\text{best}}))}{d} + 10^6 |\{\epsilon \in E \mid \text{ERT}(a, (f, d, \epsilon)) = \infty\}|,$$

where ϵ_{best} is the smallest value in E such that $\text{ERT}(a, (f, d, \epsilon)) \neq \infty$.

3.2 Benchmark Data Used

Table 1 shows the performance values of four algorithms on the benchmark functions *Sphere* and *Rastrigin separable* in 10 dimensions. The algorithm *(1+1)-CMA-ES* implements an evolutionary strategy with covariance matrix adaptation [2]. We also include two variants of Particle Swarm Optimization (PSO):

¹ <http://coco.gforge.inria.fr>

Tolerance	ERT						pERT
	10^3	10^1	10^{-1}	10^{-3}	10^{-5}	10^{-8}	
Sphere							
(1+1)-CMA-ES	1	90	267	440	611	872	87.2
PSO	1	221	2319	4911	7659	11865	1186.5
PSO-BFGS	1	68	74	74	74	75	7.5
Simplex	1	1596	3450	7283	18418	80428	8042.8
Rastrigin separable							
(1+1)-CMA-ES	4	1456464	∞	∞	∞	∞	4145646.4
PSO	4	259992	6532067	6533720	6537337	6542104	654210.4
PSO-BFGS	2	120255	∞	∞	∞	∞	4012025.5
Simplex	1	∞	∞	∞	∞	∞	5000000.1

Table 1. ERT and pERT performance values of four algorithms on the functions Sphere and Rastrigin Separable in 10 dimensions.

standard *PSO* [5] and *PSO-BFGS*, a hybrid of PSO and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [18]. The fourth algorithm is the classical *Simplex* algorithm [16,17]. Each algorithm can solve a problem instance for the Sphere function to any value of tolerance, whereas on the separable Rastrigin function this is the case only for PSO, but with higher ERT. The last column shows the pERT values that account for any failed tolerance level with a penalty.

Functions can be characterized by features. Several high-level features have been hypothesized to predict black-box optimization algorithm performance, such as multi-modality, global structure, and separability [10]. Table 2 lists the extent to which different BBOB test functions exhibit these features. *Multi-*

Function	multi-modality	global structure	separability
Sphere	none	none	high
Rastrigin separable	high	strong	high
Weierstrass	high	medium	none
Katsuura	high	none	none

Table 2. Three high-level features of four BBOB benchmark functions.

modality refers to the number of local minima of a function.

Global structure refers to the structure that emerges when only considering minima (in a minimization problem), i.e., the point set left after deleting all non-optimal points. *Separability* specifies whether a function is a Cartesian product of one-dimensional functions. For instance, the Sphere function (Fig. 1(a)) is unimodal and, therefore, has no global structure (only one point remains when restricting to minima) and is highly separable. The separable Rastrigin function (cf. Fig. 1(b)) is separable, highly multimodal, and has a strong global structure (the set of local minima forms a convex funnel).

3.3 Results

Different numerical measures have been suggested to estimate the features of a black-box function from sample evaluations. These include fitness distance correlations [13] and ICoFis, a method based on information content [14]. Furthermore, in [9] it is shown that in several cases it is possible to predict the *high-level* features of Table 2 from *low-level* features that can be computed from samples only. Based on these observations, we develop an FCA-based recommender system that suggests well-performing algorithms for a given black-box optimization problem by extracting implications from the BBOB database. The recommender system consists of three steps; cf. Algorithm 3.1. In Step 1, features of the black-box function that is given as an input are estimated by drawing samples from the function. In Step 2, these features are used to determine the features of algorithms that perform well on such functions. In Step 3, a specific algorithm is chosen that has these algorithm features. Determining good algorithm features

Algorithm 3.1 Recommender System

Input: unknown black-box function from which samples can be drawn

Output: algorithm which is suitable for optimizing input function

Step 1: estimate features of input function

Step 2: imply features of algorithms that do well on functions with such features

Step 3: pick an algorithm that has these features

(Step 2) is done using *implications* that form the background knowledge of the recommender system. The premise of an implication contains algorithm features together with features of benchmark functions, whereas the conclusion consists of attributes corresponding to the performances of the algorithms when solving the functions. The implications are obtained from the BBOB database by FCA. More precisely, we build a formal context $\mathbb{K} = (G, M, I)$, where G is the set of *objects* consisting of all pairs of benchmark functions and algorithms, M is the set of *attributes* consisting of the features of algorithms and benchmark functions together with the attributes assigned to performance values, and I is the *incidence matrix* between the objects in G and the attributes in M . We obtain the implications from the context \mathbb{K} by computing the canonical base (or another implicational base) of \mathbb{K} .

We illustrate our approach in Table 3. The objects are pairs of benchmark functions and algorithms, as introduced in Sec. 3.2. The attributes are features of algorithms, performance attributes P_i , for $i \in \{0, 5, 6, 7\}$,² and the function features (Table 2). As algorithm features we use the respective names of the algorithms together with the features *deterministic* and *hybrid* (stating that the algorithm is a combination of different search strategy). We consider Table 3 as a formal context and compute its canonical base to obtain the implications:

² To obtain a binary table, we partition the range of pERT-values into 10 intervals and introduce attribute names P_i for the intervals.

$$\begin{aligned}
& \{ (1+1)\text{-CMA-ES, multimod_high} \} \rightarrow \{ P_6 \} \\
& \quad \{ \text{PSO, sep_high} \} \rightarrow \{ P_0 \} \\
& \quad \{ \text{PSO, sep_none, multimod_high} \} \rightarrow \{ P_6 \} \\
& \{ \text{PSO-BFGS, sep_high, globstruct_strong, multimod_high} \} \rightarrow \{ P_6 \} \\
& \quad \{ \text{sep_high, globstruct_none} \} \rightarrow \{ P_0 \} \\
& \quad \{ \text{sep_none, multimod_high, globstruct_none, Alg_hybrid} \} \rightarrow \{ P_5 \} \\
& \quad \{ \text{globstruct_med} \} \rightarrow \{ P_6 \}
\end{aligned}$$

Based on this background knowledge, the recommender system would, e.g., suggest to use the algorithm PSO for functions with high separability. If additionally the function does not exhibit a global structure (e.g., the function is unimodal), then any algorithm would do well. For functions with high multi-modality, no separability, and no global structure, the system would recommend a hybrid algorithm over (1+1)-CMA-ES, confirming previous observations [12].

We note that the quality of the recommendations made by the system depends on the “quality” of the implications constituting the background knowledge. To obtain better recommendations (Step 3), suitable features of the input function need to be efficiently computable (Step 1) and the implications need to form a basis of a sufficient benchmark set (Step 2).

4 Discovering Function Features

It is not clear that the function features currently used in the optimization community define meaningful problem classes with respect to expected algorithm performance. It is an open and non-trivial problem to discover new features for this purpose. We sketch a method that could assist an expert in finding new features using *attribute exploration* from FCA [6]. This goal-oriented approach may be more practical than an undirected synthesis of new features.

In attribute exploration, we start from an initial formal context \mathbb{K} and a set \mathcal{S} of valid implications of \mathbb{K} . Then, we compute new implications $A \rightarrow B$ such that $A \rightarrow B$ is valid in \mathbb{K} , but does not follow from \mathcal{S} , where A and B are sets of attributes. In case such an implication can be found, a (human) expert has to confirm or reject $A \rightarrow B$. In the former case, $A \rightarrow B$ is added to \mathcal{S} . If the expert rejects $A \rightarrow B$, s/he has to provide a counterexample that is then added to \mathbb{K} . If no new implications can be computed any more, the search terminates.

Consider the context $\mathbb{K}_{\text{features}} = (G, M, I)$, where G is the set of benchmark functions, M the set of function features, and the incidence relation I represents the fact that a function has a certain feature. Applying attribute exploration directly to $\mathbb{K}_{\text{features}}$ would not work, as we are seeking new attributes (features), and not new objects (benchmark functions). Since objects and attributes behave symmetrically in every formal context, however, we can apply attribute exploration to the dual context $\mathbb{K}_{\text{features}}^{-1} = (M, G, I^{-1})$.

Implications from $\mathbb{K}_{\text{features}}^{-1}$ are of the form $\{ f_{i_1}, \dots, f_{i_j} \} \rightarrow \{ f_{i_{j+1}} \}$, $f_{i_k} \in G$. Presenting such implications to an expert means asking whether all features that

Testfunction	Algorithm	(1+1)-CMA-ES PSO	PSO-BFGS Simplex	no multimod. high multimod.	no gl.struct. med. gl.struct. strong gl.struct	no separ. high separ.	det. Alg. hybrid Alg.	P ₀	P ₅	P ₆	P ₇
Sphere	(1+1)-CMA-ES	✓		✓	✓			✓			
	PSO		✓	✓	✓			✓			
	PSO-BFGS			✓	✓		✓	✓			
	Simplex			✓	✓		✓	✓			
Rastrigin sep.	(1+1)-CMA-ES	✓		✓	✓	✓					✓
	PSO		✓	✓	✓	✓		✓			
	PSO-BFGS			✓	✓	✓	✓	✓			✓
	Simplex			✓	✓	✓	✓	✓			✓
Weierstrass	(1+1)-CMA-ES	✓		✓	✓	✓					✓
	PSO		✓	✓	✓	✓		✓			✓
	PSO-BFGS			✓	✓	✓	✓	✓			✓
	Simplex			✓	✓	✓	✓	✓			✓
Katsuura	(1+1)-CMA-ES	✓		✓	✓	✓					✓
	PSO		✓	✓	✓	✓		✓			✓
	PSO-BFGS			✓	✓	✓	✓	✓			✓
	Simplex			✓	✓	✓	✓	✓			✓

Table 3. Example Context.

f_{i_1}, \dots, f_{i_j} have in common are also features of function $f_{i_{j+1}}$. A counterexample would then correspond to a new feature that f_{i_1}, \dots, f_{i_j} have in common, but that is not shared by $f_{i_{j+1}}$. These are the features we are looking for.

To illustrate the method, suppose that the context $\mathbb{K}_{\text{features}}$ contains the same functions as in Table 3, but the feature *multi-modality* as the only attribute. Then, using our approach, the expert would need to judge the implication $\{\text{Rastrigin}\} \rightarrow \{\text{Weierstrass}\}$, and s/he could reject the implication by providing a “new” feature *global structure strong*, which the separable Rastrigin function has, but the Weierstrass function has not (counterexample).

5 Conclusions

We have outlined the use of FCA in building a recommender system for black-box optimization. The implications resulting from the small, illustrative example presented here are meaningful in that they both confirm known facts and suggest new ones. Attribute exploration could moreover be a powerful tool for discovering new features that are more predictive of the expected performance.

The ideas presented here are work in progress. Ongoing and future work will address the computability/decidability of features on black-box functions, and evaluate the proposed FCA-based recommender system using the BBOB database. We will investigate novel function and algorithm features, and compare the results obtained by FCA with results from clustering and regression analysis.

References

1. A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proc. of CEC-2005*, pp. 1777–1784, 2005.
2. A. Auger and N. Hansen. Benchmarking the (1+1)-CMA-ES on the BBOB-2009 function testbed. In *Proc. of GECCO-2009*, pp. 2459–2466. ACM, 2009.
3. B. Bischl, O. Mersmann, H. Trautmann, and M. Preuss. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proc. of GECCO-12*, pp. 313–320. ACM, 2012.
4. P. du Boucher-Ryan and D. Bridge. Collaborative recommending using formal concept analysis. *Knowledge-Based Systems*, 19(5):309–315, 2006.
5. M. El-Abd and M. S. Kamel. Black-box optimization benchmarking for noiseless function testbed using particle swarm optimization. In *Proc. of GECCO-2009*, pp. 2269–2274. ACM, 2009.
6. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, 1999.
7. N. Hansen, S. Finck, R. Ros, A. Auger, et al. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. 2009.
8. D. I. Ignatov, S. O. Kuznetsov, and J. Poelmans. Concept-based biclustering for internet advertisement. In *ICDM Workshop, 2012*, pp. 123–130, 2012.
9. O. Mersmann, B. Bischl, H. Trautmann, M. Preuss, C. Weihs, and G. Rudolph. Exploratory landscape analysis. In *Proc. of GECCO-11*, pp. 829–836. ACM, 2011.
10. O. Mersmann, M. Preuss, and H. Trautmann. *Benchmarking evolutionary algorithms: Towards exploratory landscape analysis*. Springer, 2010.
11. C. L. Müller, R. Ramaswamy, and I. F. Sbalzarini. Global parameter identification of stochastic reaction networks from single trajectories. In *Advances in Systems Biology*, volume 736, chapter 28, pp. 477–498. Springer, 2012.
12. C. L. Müller and I. F. Sbalzarini. A tunable real-world multi-funnel benchmark problem for evolutionary optimization (and why parallel island models might remedy the failure of CMA-ES on it). In *Proc. of IJCCI-2009*, pp. 248–253, 2009.
13. C. L. Müller and I. F. Sbalzarini. Global characterization of the CEC 2005 fitness landscapes using fitness-distance analysis. In *Proc. EvoStar*, volume 6624 of *Lecture Notes in Computer Science*, pp. 294–303. Springer, 2011.
14. M. Muñoz, M. Kirley, and S. Halgamuge. Exploratory landscape analysis of continuous space optimization problems using information content. *IEEE Trans. Evol. Comput.*, 2014.
15. M. A. Muñoz, M. Kirley, and S. K. Halgamuge. A meta-learning prediction model of algorithm performance for continuous optimization problems. In *Proc. of PPSN-2012*, pp. 226–235. Springer, 2012.
16. J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.
17. L. Pál. Comparison of multistart global optimization algorithms on the BBOB noiseless testbed. In *Proc. of GECCO-2013*, pp. 1153–1160, 2013.
18. C. Voglis, G. S. Piperagkas, K. E. Parsopoulos, D. G. Papageorgiou, and I. E. Lagaris. Mempsode: comparing particle swarm optimization and differential evolution within a hybrid memetic global optimization framework. In *Proc. of GECCO-2012*, pp. 253–260. ACM, 2012.
19. D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, 1(1):67–82, 1997.