Cell Mapping Techniques for Exploratory Landscape Analysis

Pascal Kerschke[1], Mike Preuss[1], Carlos Hernández[2], Oliver Schütze[2], Jian-Qiao Sun[3], Christian Grimme[1], Günter Rudolph[4], Bernd Bischl[5], and Heike Trautmann[1]

[1] Department of Information Systems, University of Münster, Germany
{kerschke, christian.grimme, mike.preuss, trautmann}@uni-muenster.de
[2] Department of Computer Science, CINVESTAV, Mexico City, Mexico
chernandez@computacion.cs.cinvestav.mx, schuetze@cs.cinvestav.mx
[3] School of Engineering, University of California, Merced, USA
jsun3@ucmerced.edu
[4] Department of Computer Science, TU Dortmund University, Germany
guenter.rudolph@tu-dortmund.de
[5] Statistics Department, TU Dortmund University, Germany
bischl@statistik.tu-dortmund.de

Abstract. Exploratory Landscape Analysis is an effective and sophisticated approach to characterize the properties of continuous optimization problems. The overall aim is to exploit this knowledge to give recommendations of the individually best suited algorithm for unseen optimization problems. Recent research revealed a high potential of this methodology in this respect based on a set of well-defined, computable features which only requires a quite small sample of function evaluations. In this paper, new features based on the cell mapping concept are introduced and shown to improve the existing feature set in terms of predicting expert-designed high-level properties, such as the degree of multimodality or the global structure, for 2-dimensional single objective optimization problems.

Keywords: exploratory landscape analysis, cell mapping, black-box optimization, continuous optimization, single objective optimization, algorithm selection

1 Introduction

For the optimization of difficult black-box problems, *Evolutionary Algorithms* (EA) as well as related other metaheuristics are frequently employed. However, different metaheuristics that should in principle be suitable for solving these problems often reveal enormous performance differences or do not solve some problems at all. Thus, the algorithm selection problem must be taken seriously, which means choosing the right algorithm variant and setting its parameters right. This problem has occupied numerous researchers in the last decade, see, e.g., [15] and [2] for an overview.

The *Exploratory Landscape Analysis* (ELA) approach as detailed in section 2 offers an alternative view onto algorithm analysis by focussing on problem analysis. From a small sample of evaluated search points, we calculate a set of features and learn classifiers in order to deduce what the main properties of the problem are. Then, we can make an informed guess about which algorithm should be chosen. One of the advantages of this approach is its extensibility: if new feature ideas come up, they can be seamlessly added to the existing ones. If they characterize aspects of the optimization problems well, our prediction of the problem type should improve. The only problem is that an increasing number of features will make classification more difficult. Therefore, any added feature should capture some problem aspects the standard features miss. In this work, we investigate if features obtained from cell mapping techniques (detailed in section 3) fulfill this requirement. We experimentally show in section 4 that this is indeed the case, even if only a relatively small problem sample is employed. Such a sample could be provided by random or *Latin Hypercube Design* (LHD) based initialization of a metaheuristic optimization algorithm (possibly by repeated initialization for restarts, in an amortized fashion), so that applying ELA with cell mapping features does not come at additional cost and should be used when choosing the proper algorithm for a previously unseen problem.

It should be added that while ELA focuses on real-valued optimization problems, there are related concepts for landscape analysis in combinatorial optimization, e.g., Local Optima Networks [25] for detecting the topology of different basins of attraction.

2 Exploratory Landscape Analysis

Exploratory Landscape Analysis (ELA) aims at characterizing optimization problems by means of cheaply computable features based on systematic sampling. The final goal is the construction of a model which allows for an accurate prediction of the best suited algorithm for an arbitrary optimization problem based on the computed features.

During the last years important steps into this direction for single-objective optimization problems have been made. In [18], the benchmarking framework introduced in [19], was applied to the combined experimental results of the benchmarking black-box optimization problem competition (BBOB, [10]). It was investigated whether representative algorithms exhibit similar behavior within the predefined BBOB function grouping. A set of **high-level features** derived by experts were used to characterize the functions, i.e., the degree (none, low, medium, high) of *multimodality*, *global structure*, *separability*, *variable scaling*, *search space homogeneity*, *basin-sizes*, *global to local contrast* and *plateaus*. Of course those are debatable to a certain extent. Using classification techniques based on the high-level features per function instance two clusters of algorithms could be distinguished.

In order to overcome the subjectivity of the previous approach, computable experimental **low-level features** were introduced in [17] which reflect the land-

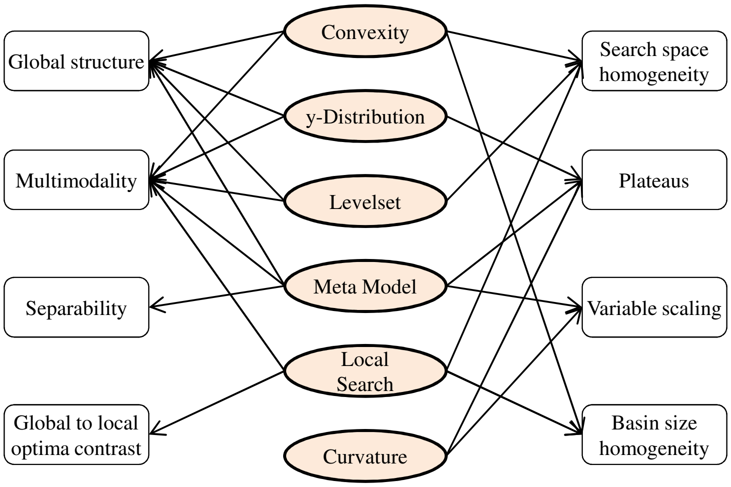


Fig. 1: Relationships between low-level (light orange) standard ELA and high-level (white) features.

scape properties of a problem. The features are grouped into six low-level feature classes, i.e., measures related to the distribution of the objective function values (*y-Distribution*), estimating meta-models such as linear or quadratic regression models on the sampled data (*Meta-Model*) and measures for the convexity and linearity (*Convexity*). Furthermore, local searches are conducted starting at the initial design points (*Local Search*), the relative position of each objective value compared to the median of all values is investigated (*Levelset*), and numerical approximations of the gradient or the Hessian represent the class of curvature features (*Curvature*). Each class contains a set of sub-features which result from the same experimental data generated from the initial data sample. Fig. 1 visualizes the assumed main relationships between the low-level feature classes and the high-level features introduced in [19].

Experimental validation of these features was conducted based on successfully predicting the values of the high-level from the corresponding low-level features. Both classification accuracy and a cost indicator representing the number of required function evaluations were included into the model building phase. Additionally, the BBOB function grouping could be perfectly predicted by specific combinations of low-level features at moderate cost. Recently, optimal algorithm selection for unseen optimization tasks was addressed in [3] by means of the BBOB09/10 results. A sophisticated cost-sensitive learning approach allowed for accurately predicting the best suited algorithm within a representative algorithm portfolio.

Additional approaches were conducted in [24], based on five features conceptually similar to [17]. In [1], new problem features categorized into the classes *problem definition, hill climbs*, and *random points* were introduced. The concept of *length scale*, which measures the ratio of changes in the objective function value to steps between points in the search space and its distribution, was suggested by [21, 22]. A first step into the direction of online algorithm selection based on low-level features is made in [23].

3 Cell Mapping

The cell mapping techniques were originally proposed by Hsu [8]. These methods are useful to determine the global behavior of nonlinear dynamical systems. The main idea of these methods is based on the fact that the representation of the numbers in a computer is finite. According to the precision of the machine, a number does not only represent the number given by its digits, but also an infinite amount of numbers within its neighborhood. The cell mapping approach employs this discretization for dividing the state space into hypercubes. The evolution of the dynamical system is then reduced to a new function, which is not defined in $\mathbb{R}^n$, but rather on the cell space.

Two cell mapping methods have been introduced in order to study the global dynamics of nonlinear systems: *simple cell mapping* (SCM), and *generalized cell mapping* (GCM). The cell mapping methods have been applied to optimal control problems of deterministic and stochastic dynamic systems [7, 9, 14]. Recently, the SCM method has been applied to multi-objective optimization [6]. For more discussions on cell mapping methods, the reader is referred to [8].

3.1 Generalized cell mapping

While the SCM offers an effective approach to investigate the global properties of a dynamical system, for problems with complicated characteristics, we need a more sophisticated algorithm. One way is to incorporate more information on dynamics of the system into the cell mapping – which is done in the GCM method. In GCM, a cell z is allowed to have several image cells, being the successors of z. Each of the image cells is assigned a fraction of the total transition probability, which is called the transition probability with respect to z.

The transition probabilities can be grouped into a transition probability matrix P of order $N_c \times N_c$, where N_c is the total number of cells. Then the evolution of the system is completely described by

$$p(n+1) = P \cdot p(n), \tag{1}$$

where p is a probability vector of dimension N_c that represents the probability function of the state. This generalized cell mapping formulation leads to absorbing Markov chains [13].

In the following, we introduce some concepts that are useful to our work.

Absorbing Markov chain A Markov chain is absorbing if it has at least one absorbing state, and it is possible to go to an absorbing state from every state (not necessarily in one step).

Classification of cells Two types of cells can be distinguished:

A *periodic cell* i is a cell that is visited infinitely often once it has been visited. In our work, we focus on periodic cells of period 1, i.e., $P_{ii} = 1$. This kind of cells correspond to the local optima candidates.

A *transient cell* is by definition a cell that is not periodic. For absorbing Markov chain, the system will leave the transient cells with probability one and will settle on an absorbing (periodic) cell.

Canonical form (cf) Consider an arbitrary absorbing Markov chain. Renumber the states so that the transient states come first. If there are r absorbing states and t transient states ($N_c = r + t$), the transition matrix has the following canonical form:

$$P = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix},$$

where Q is a t by t matrix, R is a nonzero t by r matrix, 0 is an r by t zero matrix, and I is the r by r identity matrix. Matrix Q gathers the probabilities of transitioning from some transient state to another whereas matrix R describes the probability of transitioning from some transient state to some absorbing state.

Fundamental matrix (fm) For an absorbing Markov chain the matrix $I - Q$ has an inverse $N = (I - Q)^{-1}$. The (i, j)-entry n_{ij} of the matrix N is the expected number of times the chain is in state s_j, given that it starts in state s_i. The initial state is counted if $i = j$. The matrix $fm = I + \sum_{k=1}^{\infty} Q^k$ is called the *fundamental matrix (fm)* of the Markov chain and holds the equation $fm = N$.

Absorbing probability This is defined as the probability of being absorbed in the absorbing state j when starting from transient state i, which is the (i, j)-entry of the matrix $B = NR$. In terms of cell mapping, the set of all $B_{i,j} \neq 0$ for $i \in [1, \ldots, t]$ is called the *basin of attraction* of state j, and an absorbing cell within that basin is called *attractor*.

3.2 Adaptation to Exploratory Landscape Analysis

Generalized cell mapping In the following, we assume the problem is bounded by box constraints (lb and ub), which constitutes our domain $Q = \{(x_1, \ldots, x_n)^T \in \mathbb{R}^n : \ lb_i \leq x_i \leq ub_i, \ i = 1, \ldots, n\}$. Now, we can divide each interval in N_i sections of size $h_i = (lb_i - ub_i)/N_i$. By doing this, we get a finite subdivision of the domain, where each of these elements are called regular cells. The number of regular cells is noted by N_c and we label the set of regular cells with positive integers, $1, 2, \ldots, N_c$. Also, without loss of generality, we will solely consider minimization problems.

One of the drawbacks of GCM is that in the general case, it needs more function evaluations per cell than SCM in order to find the global properties of a dynamical system. However, in the case of optimization, we can use a suitable

Algorithm 1 Construction of GCM arguments for single objective optimization

```
Require: $f$: Objective function, $s$: Set of cells
Ensure: $cf$, $fm$
  Compute the set $bc_i = \{s_j | f(s_j) < f(s_i) \text{ for all } s_j \in N_e(s_i)\}$
  Compute the set $pg_i = \{s_j | f(s_j) = f(s_i) \text{ for all } s_j \in N_e(s_i)\}$
  Compute the probability $p_{ij}$ to go from $s_i$ to $s_j$
      $$p_{ij} = \begin{cases} (f(s_i) - f(s_j)) \cdot \left( \sum_{k=1}^{|bc_i|} f(s_i) - f(s_k) \right)^{-1} & \text{, if } s_j \in bc_i \\ |pg_i|^{-1} & \text{, if } bc_i = \emptyset \text{ and } s_j \in pg_i \\ 0 & \text{, otherwise} \end{cases}$$
  Compute canonical form of $p$, $cf = \begin{bmatrix} I & 0 \\ R & Q \end{bmatrix}$
  Compute fundamental matrix of $cf$, $fm = N = (I - Q)^{-1}$
```

representative objective value $f(z)$ for each cell z, and then incorporate more information by using the comparison relationship on the set of function values.

Algorithm 1 shows the key elements to compute the characteristics needed to determine the features described within this section. For each cell z, we compare $f(z)$ to the objective values of its neighbors $N_e(z)$. Next, we assign a probability, proportional to their function values, to pass into those cells. If there is no better neighbor cell, equal transition probabilities are assigned to the neighbor cells with equal function values. Worse neighbor cells always get transition probability 0.

A key element of Algorithm 1 is the method of choosing a representative value for $f(s_j)$. In this work, we have chosen the following approaches, based on the available sample points per cell:

- *min_appr:* we select the point with the minimum objective value
- *avg_appr:* we compute the mean of all objective function values
- *near_appr:* we select the function value of the point closest to the center of the cell, even if that point is not in the cell

Please note that in case there are no points in a given cell, only the third approach is computable, whereas the other two would simply fail for these cells.

Features We now present the features that were used in order to characterize the structure of an unknown fitness landscape. In the following, we will call these the *canonical* GCM features. For each of the three approaches (min, average, closest), we consider the following features:

- *Ratio of uncertain cells* (uncert_ratio): is defined as the proportion of cells that lead to different attractors, i.e., the number of non zero entries (nnz) in matrix B at row i which are bigger than 1.

$$uncert\_ratio = \frac{1}{t} \sum_{i=1}^{t} \mathbb{1}_{[\,nnz(B_{i,1:r}) > 1\,]}$$

- *Probability to find the best cell* (prob_best): the sum of the probabilities of being absorbed by the best cell divided by the total number of cells,

$$prob\_best = \frac{1}{N_c} \sum_{i=1}^{t} B_{i,j},$$

  where j is the absorbing cell with the best function evaluation found.
- *Basin size* (bs): aggregations (standard deviation, minimum, mean and maximum) of the different basin sizes (i.e., the different colored areas in the GCM grids, cf. fig. 2 and 3).
- *Number of attractors* (attr): number of attractors (black boxes) within the grid.

Sometimes, we found considerable differences between the *min_appr* and *avg_appr* approach. In order to study this, we consider the following three features:

- *Common periodic cells* (common.pcells): number of common periodic cells between the approaches. Let $pcell_{avg}$ and $pcell_{min}$ be the periodic cells of the *avg_appr* and *min_appr* approaches respectively, then

$$common.pcells = \left| pcell_{avg} \bigcap pcell_{min} \right| .$$

- *Common transient cells* (common.tcells): number of common transient cells between approaches. Let $tcell_{avg}$ and $tcell_{min}$ be the transient cells of the *avg_appr* and *min_appr* approaches respectively, then

$$common.tcells = \left| tcell_{avg} \bigcap tcell_{min} \right| .$$

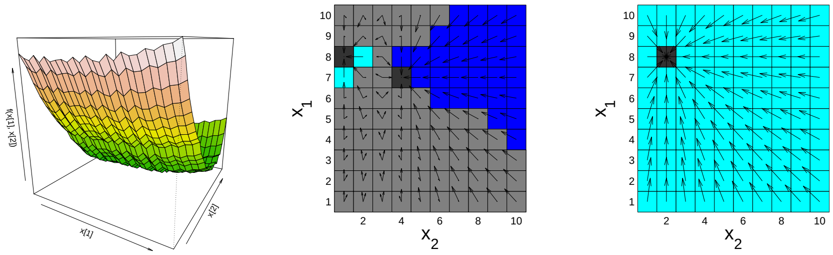
- *Percentage of different cells* (common.dcells): the percentage of cells which change their roles (absorbing and transient) from one approach to the other one.

$$common.dcells = 1 - \frac{1}{N_c}(common.pcells + common.tcells)$$

3.3 Examples

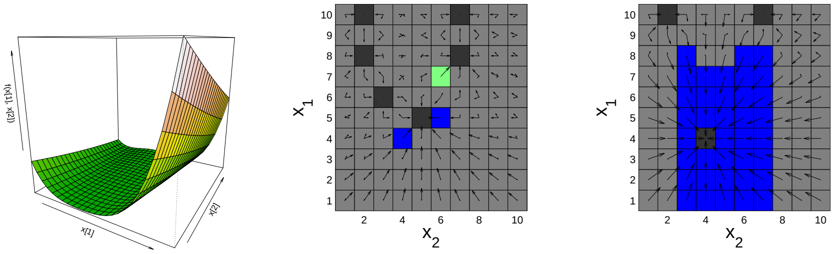
In the following, we present two examples using the GCM approach on a 10×10 grid on functions, taken from the BBOB benchmark suite [10]. We employ two approaches for choosing the representative objective function value of each cell: the minimum of each cell (*min_appr*) and the average (*avg_appr*) of the function evaluations.

In all figures, dark gray cells represent an attractor, light gray cells are cells that belong to more than one basin of attraction. All other colors refer to a different basin of attraction, and the arrows represent the mappings from one cell to another.



(a) graph (b) min_appr (c) avg_appr

Fig. 2: Rastrigin (BBOB-ID 3) with GCM approach



(a) graph (b) min_appr (c) avg_appr

Fig. 3: Rosenbrock (BBOB-ID 8) with GCM approach

For the Rosenbrock function (fig. 3), both approaches show different characteristics. We can observe that the numbers and locations of attractors are different as well as the sizes of the basins of attraction. In the *min_appr* approach, the parabolic shaped flat valley is reflected by the u-shaped locations of the attractors. For the Rastrigin problem (fig. 2), the *avg_appr* reveals the underlying global structure whereas the *min_appr* provides the main basin of attraction. For the simple, unimodal sphere problem (not shown here), both approaches look like the *avg_appr* of the Rastrigin problem.

3.4 Additional Cell Based Features

Taking the canonical GCM features in section 3.2 into account, one may look for even more features that use the overall idea of GCM, namely discretization. From [20], we know that the most important high-level ELA features for the partition of the 24 BBOB problems into different groups are *multimodality*, *variable scaling*, and *global structure*. It appears difficult to recognize the *variable scaling* (the deformation of basins due to extreme gradient differences in orthogonal directions) with only a relatively small and evenly distributed sample at hand,

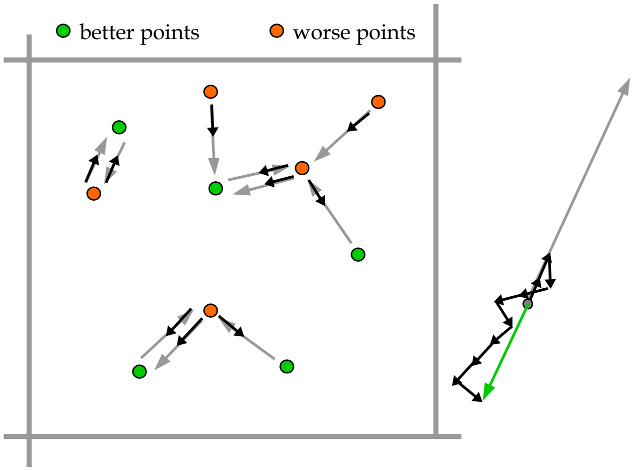


Fig. 4: Schematic view of the *gradient homogeneity* features: for every point, the vector to the nearest point is determined (gray) and normalized so that it points to the better point (black). All black vectors are added (green) and the length of the resulting vector is compared to the added length of all vectors (right).

because it would be necessary to place multiple points in close vicinity to their neighbors for estimating gradients in different directions.

However, the discretization of a sample into a number of cells, with several points in each cell, opens up possibilities to measure *global structure* and *multimodality*. Note that all features we suggest in this section, except for the last group (*convexity*), are independent of the search space dimensionality. The only precondition to computing them is that we have on average more than one point in each cell. The reason for the independence is that interactions between different cells are ignored, we focus on the cell contents and aggregate the values computed per cell over all cells. The features defined in section 3.2 may also be transferable to higher dimensions, but this would not be trivial. In the following, we discuss the obtained features in groups that each follow a different concept.

- *Gradient homogeneity* (gradhomo): Fig. 4 visualizes the general idea. For every point within a cell's sample, we find the nearest neighbor and compute the individual, normalized difference vector, which is always rotated so that it is pointing to the worse point. Then, we compute the length of the vector sum of the individual vectors amd divide it by the maximal possible vector length (equals the number of points due to normalization). In the figure, we obtain a value in the range of 0.5, which reflects well that there is a trend for better points towards the bottom of the cell, but there is also some *multimodality*. For completely randomly distributed objective values, the fraction should be around 0 (vectors pointing in all directions), for a strong trend the values should approach 1.0 (all vectors point into the same direction). This is conceptually close to simple step-size adaptation heuristics

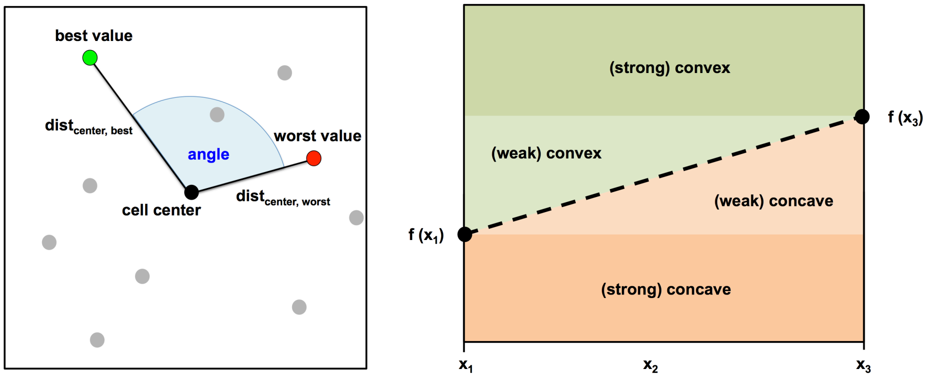


Fig. 5: The length of the vectors from the center to the best and worst value within a cell, as well as the angle between those vectors summarize the direction of the landscape (left). Comparing three (horizontally, vertically or diagonally) neighbouring cells allows to draw conclusions on the local convexity (right).

for the CMA-ES as discussed in [12]. From the individual values for each cell, we obtain two features by computing the mean and the standard deviation over all cells. Note that we ignore vector direction differences between cells, only the homogeneity within each cell is taken into account. Simple unimodal functions shall thus generate very high mean values.

- *angle*, *dist_best* and *dist_worst*: Motivated from the previous feature, the location of the best and worst values within the cell might return some insight of the landscape (cf. fig. 5). If they lie in opposite directions it indicates a trend within the cell. In that case the angle between the vectors from cell center to worst value and cell center to best value would be close to 180°. Two features are obtained by aggregating the angles of all cells from the grid using the mean and the standard deviation. Furthermore, the standard deviations in the lengths of the two vectors are used as additional features. In case of simple functions as the *sphere* function, the variation should be low as the majority of the cells have similar distances, because they usually lie close to the borders of the cells. In very multimodal functions, the variation should be high as cells with local optima result in contrary distances (short distances of the best values and long distances of the worst values) compared to cells without any local optima.
- *fun_ratio*: Using the best and worst values provides further information. So far, the features only used the location within the decision space, but their function values were disregarded. Using them, two more features can be obtained. We compute the mean and standard deviation of the distances between the best and worst function values within a cell, scaled by the distance of best and worst function value within the entire grid.
- *convex_weak*, *convex_strong*, *concave_weak* and *concave_strong*: These four features focus on the convexity of the functions' landscape. For any three (in

a line) neighboring cells, the observations, which are located closest to the cell centers (x_1, x_2 and x_3), should also be more or less in a line. A function is said to be (weak) convex, if $f(\alpha \cdot x_1 + (1-\alpha) \cdot x_3) > \alpha \cdot f(x_1) + (1-\alpha) \cdot f(x_3)$ for $\alpha \in (0,1)$. Assuming that x_2 lies approximately in the middle of x_1 and x_3, i.e., $\alpha = 0.5$, the function is convex if $f(x_2) > 0.5 \cdot (f(x_1) + f(x_3))$. Furthermore, it is strong convex, if $f(x_2) > \max\{f(x_1), f(x_3)\}$. The concavity can be derived analogously (cf. fig. 5). Based on that approach, each of the four features can be derived by computing its ratio over all possible combinations of three (either horizontally, vertically or diagonally) neighboring cells within the grid.

4 Experimental Analysis

Our overall goal is to identify the features that enable predicting the high-level properties of an unknown problem in order to select a matching optimization algorithm. In the ideal case, we would aim at a diverse set of decision space dimensions so that we obtain a universally valid classifier. However, the canonical GCM features of section 3.2 can only be computed for 2D problems without a sophisticated redesign of cell location and neighborhoods. We therefore restrict this first analysis to 2D. Although we assume that this setting should be easier than 5D or 10D as attempted on the 24 BBOB functions in [17], we currently have no comparison data available as in that work, the easier case of leave-one-instance-out cross-validation was considered. Thus, our comparison will be a relative one between the standard ELA features, the whole set of new features of section 3.2 and 3.4, and all of these combined. We employ feature forward selection to find well-performing, but small feature sets for each feature group. We aim at detecting for which high-level features (as *multimodality*) the new features actually provide a considerable advantage. The experiments were run using `MATLAB` [16] (canonical GCM features) and `R` [26] (additional GCM features), for resampling and feature selection the `R` package `mlr` [5] was employed.

4.1 Experimental Setup

A sample of 1000 evaluations, randomly distributed over $10 \times 10 = 100$ cells in 2D was employed for all considered problems. This is a relatively small number, but still larger than the number of initial samples used, e.g., in most EAs. However, for difficult multimodal problems, 1000 is small in comparison to the number of function evaluations necessary to solve them. Additionally, the surplus from the setup of a start population could be used for the initialization of restarts so that not too many evaluations are wasted.

The experiments are based on the 24 BBOB functions, for each one we select 10 function instances and perform 10 statistical replications. The features were averaged over the replications, providing a reduction of the variance among the stochastic features. Thus, the setup consists of a total of 240 instances. The 50 low-level ELA features, introduced in [19] were reduced to 22, as the feature

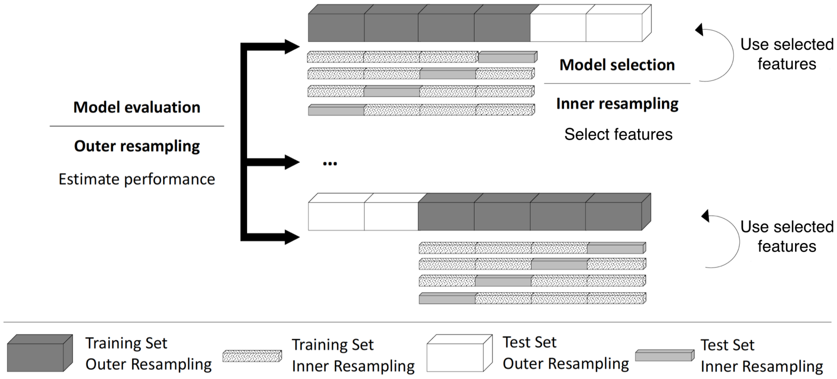


Fig. 6: Nested Resampling strategy for feature selection, inspired by [4].

groups belonging to local search, convexity and curvature were discarded due to their need for additional function evaluations. In addition to those 22 ELA features, 44 GCM features were used: two common cells features, three approaches covering ten features each (cf. section 3.2), and the additional cell based features (cf. section 3.4). Each high-level property will be predicted by a *random forest* classifier (using default settings, i.e., 500 trees [11]). Missing values among any of those 66 features were imputed with a value twice as high as the highest non-missing value within that feature, which is a reasonable and standard imputation technique for tree-based methods.

The model validation [4] is done using a nested resampling strategy as visualized in fig. 6, which is a standard evaluation approach in machine learning for feature selection scenarios as ours. In order to generate a realistic scenario, the functions were blocked for the modeling, i.e., all instances that belong to the same BBOB function were used either for training or testing. This way, the data can be split up into a maximum of 24 blocks – one per function. Both, the inner and outer loops, use a *leave-one-function-out* (LOFO) *cross-validation* (CV). Thus, the outer loop partitions the data into 24 blocks, each one consisting of one BBOB function (10 instances, colored white in fig. 6) in the test data and the remaining 23 functions (230 instances, dark gray) in the corresponding model selection set. On each of the model selection sets, forward selection is used for selecting the best feature sets. To evaluate a potential feature set, the random forest performance on this feature set is calculated using a LOFO CV (in the inner loop) on the 230 instances of the model selection set. When the feature forward selection has terminated, a random forest is finally trained on the whole model selection set with the selected feature set and its *misclassification error* (MCE) is measured on the corresponding test data. Thus, the resampling strategy returns 24 unbiased performance values and feature sets (one per fold of the outer loop).

It is important to understand that blocking the functions and using the nested resampling approach leads to a more realistic estimation of the MCE as

Function	multim.	gl.-struc.	separ.	scaling	homog.	basins	gl.-loc.
1: Sphere	none	none	high	none	high	none	none
2: Ellipsoidal separable	none	none	high	high	high	none	none
3: Rastrigin separable	high	strong	high	low	high	low	yes
4: Büche-Rastrigin	high	strong	high	low	high	med.	yes
5: Linear Slope	none	none	high	none	high	none	none
6: Attractive Sector	none	none	none	low	med.	none	none
7: Step Ellipsoidal	none	none	none	low	high	none	none
8: Rosenbrock	low	none	none	none	med.	low	yes
9: Rosenbrock rotated	low	none	none	none	med.	low	yes
10: Ellipsoidal high conditioned	none	none	none	high	high	none	none
11: Discus	none	none	none	high	high	none	none
12: Bent Cigar	none	none	none	high	high	none	none
13: Sharp Ridge	none	none	none	low	med.	none	none
14: Different Powers	none	none	none	low	med.	none	none
15: Rastrigin multimodal	high	strong	none	low	high	low	yes
16: Weierstrass	high	med.	none	med.	high	med.	yes
17: Schaffer F7	high	med.	none	low	med.	med.	yes
18: Schaffer F7 moderately ill-cond.	high	med.	none	high	med.	med.	yes
19: Griewank-Rosenbrock	high	strong	none	none	high	low	yes
20: Schwefel	low	none	none	none	high	low	yes
21: Gallagher 101 Peaks	low	none	none	med.	high	med.	yes
22: Gallagher 21 Peaks	low	none	none	med.	high	med.	yes
23: Katsuura	high	none	none	none	high	low	yes
24: Lunacek bi-Rastrigin	high	none	none	low	high	low	yes

Table 1: Classification of the BBOB functions based on their properties (*multimodality*, *global structure*, *separability*, *variable scaling*, *homogeneity*, *basin sizes*, *global-to-local*). Predefined groups are separated by horizontal lines and changes to previous versions are colored red.

this approach avoids overfitting to the training data. The approach as a whole is different from the one used in [17] and should lead to a less optimistic, but more realistic classification quality assessment.

In order to handle very small classes, which lead to problems during (blocked) cross-validation, some classes within the properties *multimodality* (low and medium), *global structure* (deceptive, weak and none) and *global-to-local* (low, medium and high) were merged. The property *plateau* was removed completely as it was a 2-class problem, with one class consisting of only one observation. All used properties are shown in table 1.

4.2 Results and Discussion

Comparing the three performance values (one per feature group, i.e., ELA, GCM and their combination) for each of the seven high-level properties shows, that the GCM features improve the ELA features in five of the seven categories. Table 2 reveals that especially the properties *global structure*, *homogeneity* and *multimodality* benefit from the addition of the GCM features as the corresponding *mean misclassification errors* (MMCEs), i.e., the mean over the MCEs of the 24 folds, were reduced by 10-20%. As the BBOB set was created in a way that

property	**Median**			**Mean**		
	all	**ELA**	**GCM**	**all**	**ELA**	**GCM**
Basin Size	0.40	**0.20**	0.35	0.47	**0.31**	0.47
Global to Local	**0.00**	**0.00**	**0.00**	**0.14**	0.16	0.19
Global Structure	**0.00**	0.10	**0.00**	**0.18**	0.34	0.21
Homogeneity	**0.00**	0.35	0.20	**0.28**	0.39	0.34
Multimodality	**0.00**	0.35	**0.00**	**0.15**	0.36	0.20
Separability	**0.00**	**0.00**	**0.00**	**0.17**	0.20	0.24
Variable Scaling	0.25	**0.00**	0.60	0.39	**0.28**	0.53

Table 2: Comparison of the MCEs, aggregated using the median and mean over the 24 folds of the outer LOFO CV (best performances written in bold type). Based on a Wilcoxon signed-rank test, the differences between all features and the ELA features were significant for the properties *global structure* and *multimodality* (w.r.t. significance niveau 10%). Also, there were significant differences between ELA and GCM for *multimodality* and *variable scaling*.

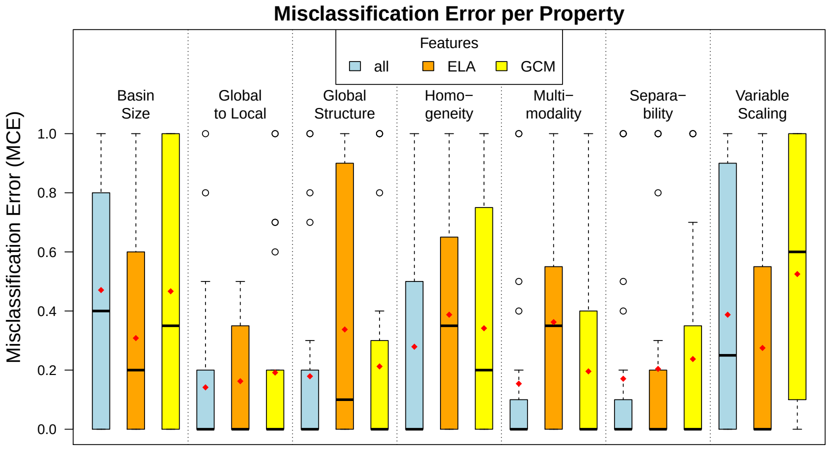
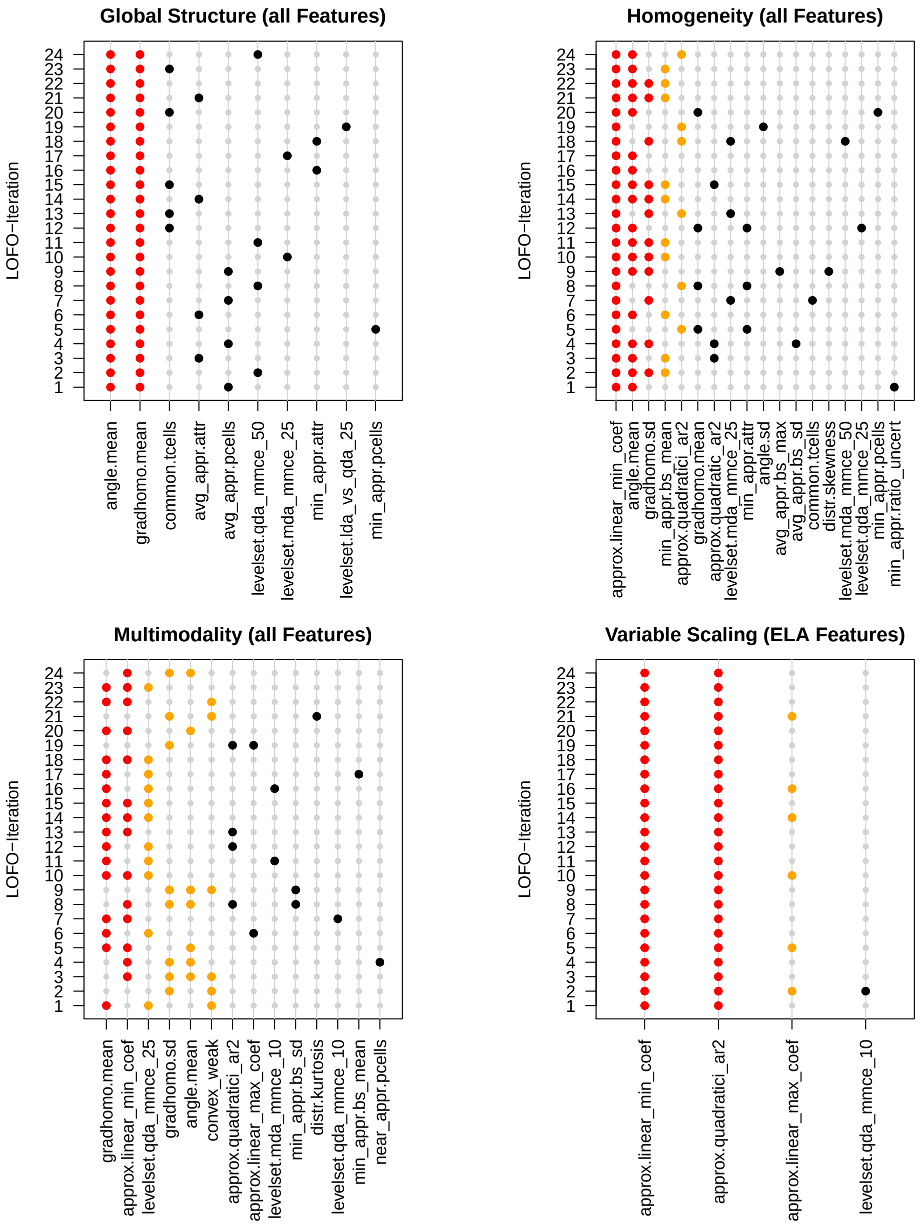


Fig. 7: Boxplots of MCEs per property and feature subset. Each boxplot is based on 24 performance values, obtained during the model evaluation. The red diamonds indicate the mean of each sample, i.e., the MMCE.

each function of that set covers different aspects, the variance within the misclassification rates is quite high. However, using a Wilcoxon ranked-sum test, it could be shown that the improvements in *global structure* and *multimodality* are statistically significant w.r.t. a significance niveau of 10%, which is remarkable considering the few performance values. Given that none of the GCM features explicitly aims at explaining properties like *variable scaling*, it is very reasonable that the new features were not able to improve the performance of this property. Instead, the performance decreased significantly, probably due to adding redundant features.


Global Structure (all Features)

Homogeneity (all Features)

Multimodality (all Features)

Variable Scaling (ELA Features)

Fig. 8: The figures above show the selected features per fold within the model validation. All the illustrated cases reveal major differences between the feature groups. The color indicates whether the feature was chosen at least once (black), in at least 25% (orange), or in at least 50% (red) of the folds.

It is also noteworthy that the MMCE of four properties is below 20%, which is good w.r.t. the fact that a very strict and realistic validation method (nested resampling with leave-one-function-out cross-validations) was applied.

As mentioned before, each BBOB problem describes a different problem and thus, their characteristics are very diverse. Hence, it is reasonable to compare all performances, e.g., using boxplots (cf. fig. 7), instead of comparing aggregated measurements such as median or mean. Comparing the performances over all 24 folds also reveals that the MCEs are skewed positively, i.e., in the majority of iterations the models are very good and therefore fail only in a few iterations.

Furthermore, one might be interested in the selected features. Due to the nested resampling strategy, 24 (different) feature sets exist, which cannot be aggregated. Instead, it is more reasonable to look at the importance of the selected features, e.g., by analyzing how often each feature was selected. Fig. 8 shows the importance plot for the four cases in which ELA differed strongly from the other feature sets. In matters of *global structure*, the *angle* and *gradient homogeneity* features were selected in each of the 24 subsets and therefore they seem to be the features which mainly describe this property. Also, both of these features are, combined with two *meta model* features (ELA) and another GCM feature, important for explaining the *homogeneity*. Adding those two features towards some *meta model* and *levelset* features also leads to a major improvement in describing the *multimodality* of a function. However, in case of *variable scaling* the ELA features, especially three features from the *meta model* group, provide already sufficient information, which deteriorated by the perturbation of the significantly worse GCM features.

5 Conclusions and Outlook

We have approached the extension of the standard ELA feature set from the perspective of discretization, namely by using *general cell mapping* (GCM) properties as features in order to better predict the high-level properties as *multimodality* and *homogeneity*. Furthermore, we have extended the canonical GCM features by a set of newly designed features that use only the assignment of observations (search points) to cells and are therefore (with exception of the *convexity* features) completely independent of cell location and neighborhood, and thus of the number of dimensions. Whereas it would be nontrivial to extend the canonical GCM features to more than 2D, this requires no change other than the redefinition of cells for the additional features of section 3.4.

For the aforementioned reasons, the experimental analysis focused on 2D with a relatively small sample of 1000 points. The results show that the new features are especially valuable for predicting the high-level properties *multimodality* and *global structure*, which are, according to the original ELA experiments, most important for selecting a proper algorithm for a difficult black-box problem. Especially the new (additional) *angle* and *gradient homogeneity* features are chosen regularly by the feature selection, whereas the canonical GCM features play only a minor role.

This is not only a very good improvement but also reveals how other successful features should be created. As the ELA approach can easily integrate new features, there are endless possibilities for designing features in order to improve the classification even for the 2 (of 7) high-level properties that have not been improved. Additionally, the feature selection process itself shall be investigated and improved further (if feature selection would be perfect, i.e., if finding the best subset of features would be guaranteed, adding more features could never result in deterioration). Simple forward selection is obviously not ideal, but total enumeration is also not possible due to the combinatorial explosion. One shall try more clever heuristics or meta-heuristics such as EAs for further improvement.

Acknowledgements

This work was supported by the grant ANR-12-MONU-0009-03 (NumBBO) of the French National Research Agency. The third author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN.

References

1. Abell, T., Malitsky, Y., Tierney, K.: Features for exploiting black-box optimization problem structure. In: Nicosia, G., Pardalos, P.M. (eds.) LION. Lecture Notes in Computer Science, vol. 7997, pp. 30–36. Springer (2013)
2. Bartz-Beielstein, T., Chiarandini, M., Paquete, L., Preuss, M.: Experimental methods for the analysis of optimization algorithms. Springer (2010)
3. Bischl, B., Mersmann, O., Trautmann, H., Preuss, M.: Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In: Proceedings of the 14th annual conference on Genetic and evolutionary computation. pp. 313–320. GECCO '12, ACM, New York, NY, USA (2012)
4. Bischl, B., Mersmann, O., Trautmann, H., Weihs, C.: Resampling methods in model validation. Evolutionary Computation Journal 20(2), 249–275 (2012)
5. Bischl, B.: mlr: Machine Learning in R., https://github.com/berndbischl/mlr, R package version 1.2
6. C. Hernández and Y. Naranjani and Y. Sardahi and W. Liang and O. Schütze and J.-Q. Sun: Simple Cell Mapping Method for Multi-objective Optimal Feedback Control Design. International Journal of Dynamics and Control 1(3), 231–238 (2013)
7. C. S. Hsu: A discrete method of optimal control based upon the cell state space concept. Journal of Optimization Theory and Applications 46(4), 547–569 (1985)
8. C.S. Hsu: Cell-to-cell mapping: A method of global analysis for nonlinear systems. Applied mathematical sciences, Springer-Verlag (1987)
9. F. H. Bursal and C. S. Hsu: Application of a cell-mapping method to optimal control problems. International Journal of Control 49(5), 1505–1522 (1989)
10. Hansen, N., Auger, A., Finck, S., Ros, R.: Real-parameter black-box optimization benchmarking 2010: Experimental setup. Tech. Rep. RR-7215, INRIA (2010), http://hal.inria.fr/docs/00/46/24/81/PDF/RR-7215.pdf
11. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer, 2 edn. (2009)

12. Jägersküpper, J., Preuss, M.: Empirical investigation of simplified step-size control in metaheuristics with a view to theory. In: McGeoch, C.C. (ed.) Experimental Algorithms, 7th International Workshop, WEA 2008. Lecture Notes in Computer Science, vol. 5038, pp. 263–274. Springer (2008)
13. Kemeny, J., Snell, J.: Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix". Undergraduate Texts in Mathematics, Springer (1976)
14. L. G. Crespo and J. Q. Sun: Stochastic Optimal Control of Nonlinear Dynamic Systems via Bellman's Principle and Cell Mapping. Automatica 39(12), 2109–2114 (2003)
15. Lobo, F.G., Lima, C.F., Michalewicz, Z. (eds.): Parameter Setting in Evolutionary Algorithms, Studies in Computational Intelligence, vol. 54. Springer (2007)
16. MATLAB: version 8.2.0.701 (R2013b). The MathWorks Inc., Natick, Massachusetts (2013)
17. Mersmann, O., Bischl, B., Trautmann, H., Preuss, M., Weihs, C., Rudolph, G.: Exploratory landscape analysis. In: Proceedings of the 13th annual conference on Genetic and evolutionary computation. pp. 829–836. GECCO '11, ACM, New York, NY, USA (2011)
18. Mersmann, O., Preuss, M., Trautmann, H., Bischl, B., Weihs, C.: Analyzing the BBOB Results by Means of Benchmarking Concepts. Evolutionary Computation Journal (2014, accepted for publication)
19. Mersmann, O., Trautmann, H., Naujoks, B., Weihs, C.: Benchmarking evolutionary multiobjective optimization algorithms. In: Ishibuchi, H., et al. (eds.) IEEE Congress on Evolutionary Computation (CEC). pp. 1311–1318. IEEE Press, Piscataway NJ (2010)
20. Mersmann, O., Preuss, M., Trautmann, H.: Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In: Schaefer, R., Cotta, C., Koodziej, J., Rudolph, G. (eds.) Parallel Problem Solving from Nature, PPSN XI, Lecture Notes in Computer Science, vol. 6238, pp. 73–82. Springer Berlin Heidelberg (2010)
21. Morgan, R., Gallagher, M.: Length scale for characterising continuous optimization problems. In: Coello, C.A.C., et al. (eds.) Parallel Problem Solving from Nature - PPSN XII, Proceedings, Part I. LNCS, vol. 7491, pp. 407–416. Springer (2012)
22. Morgan, R., Gallagher, M.: Using landscape topology to compare continuous metaheuristics: A framework and case study on edas and ridge structure. Evol. Comput. 20(2), 277–299 (Jun 2012)
23. Muñoz, M.A., Kirley, M., Halgamuge, S.K.: Landscape characterization of numerical optimization problems using biased scattered data. In: IEEE Congress on Evolutionary Computation. pp. 1–8. IEEE (2012)
24. Muñoz, M.A., Kirley, M., Halgamuge, S.K.: A meta-learning prediction model of algorithm performance for continuous optimization problems. In: Coello, C.A.C., et al. (eds.) Parallel Problem Solving from Nature - PPSN XII, Proceedings, Part I. LNCS, vol. 7491, pp. 226–235. Springer (2012)
25. Ochoa, G., Tomassini, M., Vérel, S., Darabos, C.: A study of nk landscapes' basins and local optima networks. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation. pp. 555–562. GECCO '08, ACM, New York, NY, USA (2008), http://doi.acm.org/10.1145/1389095.1389204
26. R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2013), http://www.R-project.org/