Niching Foundations: Basin Identification on Fixed-Property Generated Landscapes

Mike Preuss
TU Dortmund
Germany
mike.preuss@tu-dortmund.de

Catalin Stoean
University of Craiova
Romania
catalin.stoean@inf.central.ucv.ro

Ruxandra Stoean
University of Craiova
Romania
ruxandra.stoean@inf.central.ucv.ro

ABSTRACT

The performance of niching based or related evolutionary algorithms clearly depends on problem properties as e.g. the number of local optima of a problem. We assume there must be more such properties currently not taken into account and, following from practical experience, suggest two more, namely *basin size contrast* (*BSC*), the size relation of the largest and the smallest basin, and *global to local optima contrast* (*GLC*), the height relation of the global and an average local optimum. We investigate the effect of these problem properties on the performance of different basin identification methods (as subtasks of niching algorithms), namely nearest-better clustering, detect-multimodal, and Jarvis-Patrick clustering, individually, or in combinations. Employing an existing problem generator that enables complete control and knowledge of basins, instances are generated and validated according to predefined property values and the basin identification performance data is modeled in order to detect similarities that may be interpreted as effects of the stated properties. We also give recommendations concerning usage of basin identification methods in different situations. Our approach is strongly related to the recently suggested general idea of *exploratory landscape analysis* (ELA).

Categories and Subject Descriptors

G.1.6 [**Mathematics of Computing**]: Optimization—*Global Optimization, Unconstrained Optimization*; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*

General Terms

Experimentation, Measurement, Performance

Keywords

exploratory landscape analysis, clustering, global optimization, basin identification, niching

GECCO'11, July 12–16, 2011, Dublin, Ireland.

1. INTRODUCTION

When optimizing multimodal blackbox problems, *evolutionary algorithms* (EAs) and also other heuristic or deterministic optimization methods are prone to sooner or later end up in a single basin of attraction which is not necessarily the one carrying the global optimum. This can only be cured by organizing search in a way that the EA operates on different search space areas separately (sequentially or in parallel). This is the fundamental idea of niching algorithms, and the most developed EAs for real-valued optimization (e.g. the CMA-ES presented in [8]) also adhere to this principle, although they do not employ spatial separation. Despite the fact that parallelism is an interesting option for implementing separate searches, we completely disregard the timing issue in this work and concentrate on the space issue only: Where shall the restarts take place? Basically, two different approaches are possible.

- Performing restarts in random places, e.g. with different algorithm parameters. The recent CMA-ES variants IPOP-CMA-ES [3] and BIPOP-CMA-ES [6] are good examples for this simple strategy. Random restarts require no knowledge of the optimized function and shall be the used if obtaining such knowledge proves difficult.

- Dissecting the search space into possible basins of attraction according to the data collected online and let the optimization method work on the basins separately. This approach is followed by niching algorithms, and there is a plethora of methods designed for this purpose, possibly starting with crowding and fitness sharing.

In the following, we concentrate on the niching approach, although it is more complex and not suitable for all multimodal problems. E.g., it may be argued that methods relying on distance measuring will not work well in very high dimensions, as for $D \gg 15$, the contrast between measures dramatically drops [4], and for $D \to \infty$, points in search space are effectively all equally distant from each other. However, there are many engineering problems with moderate dimensionality between 5 and 20, where niching should work well. In the past, many experimental investigations of niching methods have been performed to a large extent on 1D, 2D and 3D functions, which has been heavily criticised due to lack of practical relevance. Interestingly, analysing the results of the BBOB'09 competition [7][1] shows that behavior of the optimization algorithms on 5 to 20 dimensions is relatively consistent [10]. We deduce that this is at least partly caused by the problems which

[1]results: `http://coco.gforge.inria.fr/doku.php?id=bbob-2009-results`, this site also contains the BBOB'10 results which are more complete but qualitatively similar

behave rather similar when the dimension is changed in this range. For these reasons, we confine our investigation to 5D problems instances only, seemingly the most simple ones with practical relevance and still suitable for niching methods. Note that we do not advocate niching methods as the only possible optimization techniques for complicated real-world problems, but if niching methods shall be successfully applied, it is important to answer the *where is elsewhere?* question posed by [21] in order to find meaningful starting places for separate populations.

Among niching methods, we may make the distinction between implicit and explicit basin identification. The latter means that we learn where the basins are, which requires to establish a matching between sampled search points and different basins. Optimization algorithms may then use this knowledge to start separate parallel strands of local optimization. Such methods realize the niching paradigm as stated in [12], adapted from versions defined by other authors: *Niching in EAs is a two-step procedure that a) concurrently or subsequently distributes individuals onto distinct basins of attraction and b) facilitates approximation of the corresponding (local) optimizers.*

Implicit basin identification means that the current population distributes over different basins, but we do not know the matching (arguably, this is the case for standard crowding and fitness sharing). The EA may still successfully form separate populations and thus speed up search, but the algorithm (and the user) is not able to extract any problem knowledge from the runs. This in turn complicates explaining algorithm performance, as it is difficult to name reasons for success or failure when it is to a large extent unclear how algorithm and problem interact.

In the following, we start by envisioning some problem properties and show how to create instances that respect these properties, using an already existing problem generator (section 2). After introducing the compared basin identification methods in section 3, we then perform an experimental study that employs these basin identification mechanisms and combinations thereof in order to find relations between their performance (and cost) and the given properties. This is done in two stages, on randomly generated initial sample data, and on already developed populations that could occur after running a few generations (section 4).

We pursue two research questions, namely how the developed problem properties affect the difficulty of an otherwise fixed problem, and which of the tested basin identification methods is best suited to extract basin information out of these problems. This is novel in different ways, and in some sense more low-level than other approaches as we do not compare complete algorithms (as has e.g. been done in [17]) but the base mechanisms of niching methods. The two problem properties employed for building test problems are suggested in [13] but have never been investigated by themselves. [10] found that their role in explaining algorithm performance on the BBOB test set is limited (where the number of multimodal problems without global structure is very small), but practical experience and reports from others (e.g. [19]) tells us that they a) have some relevance for the performance of algorithms and b) will also occur in real-world problems. Of course, with showing different performances of basin identification methods on our designed benchmark problems we cannot prove that it is these properties that make algorithms better or worse on specific multimodal problems, neither can we safely assume that they are important features of real-world problems. However, observed performance differences hint at the possibility that both is the case. In the absence of a more complete theory, this is a step towards explaining what actually makes global optimization problems difficult to solve for stochastic search algorithms and what we can do about it.

2. GENERATING PROBLEMS WITH DISTINCTIVE PROPERTIES

When investigating optimization methods on multimodal problems, we can imagine many problem features which influence performance. Unfortunately, there is no good understanding of the hardnesses imposed by different features at the moment, maybe with some exceptions[2]: number of local optima, global structure in location of optima and separability.

Looking for one global optimum is of course harder if the number of misleading peaks grow, but if the peaks follow a global scheme as for the Rastrigin function or funnel problems, this property may be used to increase the search speed again. In the BBOB'09 benchmark problem set [7], most multimodal problems have therefore been labeled as possessing strong or weak global structure, and observations on real-world problems show that problems with strong global structure indeed exist, e.g. [1].

A separable problem is easier to solve than one with interacting dimensions because it can be transformed into several one-dimensional subproblems. However, the named properties only allow for a very rough classification of optimization problems, more of them could surely be found/defined and exploited by optimization methods. Eventually, establishing more problem properties should be useful also for optimizing real-world problems, even if one usually lacks the domain knowledge to sort any problem into a classification in an ad-hoc way. Dealing with a real-world problem is usually an iterative process of algorithm adaptation and learning (from results), and being aware of possible problem properties and their relation to the used algorithms surely helps in setting up a good optimization algorithm faster. Defining some properties would thus enable some kind of *explorative landscape analysis* (ELA, [13] and [10]) which only means that the currently often very informal process of iterative adaptation of algorithms could be replaced by a somewhat more structured approach.

For now, we resolve to employing our intuitive understanding of what kind of problem structure makes optimization easier or harder and set up some properties we expect to be meaningful. We then use a problem generator to create instances with the desired property values. It is important to note that the concrete generator itself plays a minor role and could easily be exchanged, but it must allow for complete knowledge of the basin locations in order to check the output of basin identification methods investigated later on.

2.1 Landscape Properties

Next to the already named properties, we concentrate on two more here, namely *basin size contrast* and *global to local optima contrast*. Further properties surely make sense, but are disregarded in this study.

Basin size contrast [19] already stated that the basin size of the global optimum certainly influences the hardness of a problem. We argue that an optimization algorithm dealing with a multimodal problem shall try to locate many optima, and, as it is not known a priori which is the global one, the size relation of the largest to the smallest basin should give a meaningful measure of problem hardness.

$$\mathrm{bsc}(f) := \frac{\min(V(b_i))}{\max(V(b_i))} \quad \text{for basins } b_i \text{ of function } f, V = \text{volume} \tag{1}$$

A small *BSC* means a big difference and thus a tendency towards a harder problem (if the number of peaks is not very small), a large

[2]besides dimensionality, which usually positively correlates to problem hardness and is always known prior to optimization

BSC (approaching 1) stands for a low difference in basin sizes and a rather simple problem.

Note that many algorithms for multimodal problems (e.g. most niching EA methods) assume similar basin sizes and use appropriately adjusted distances for differentiating between basins. If size differences are huge, these methods are doomed to fail. Some algorithms therefore attempt to adapt to basin sizes which should give a clear advantage in these cases.

Global to local optima contrast refers to the height (objective function) differences between global and local peaks in comparison to the average fitness level of a problem. It thus determines if the best peaks are easily recognized as such because their relative fitness distance to average function values is much larger than the one for average local optima. [5] give an example for a bit-coded problem where the difference between the best peak and the other peaks is explicitly controlled (modified P-PEAKS generator). The Katsuura function (no. 23) as employed in the BBOB'09 test set [7] is an example of a problem with a low difference between the global and local peaks, compared to average function values, and, according to BBOB'09 and BBOB'10 results ([2]), is is practically unsolvable for all tested methods. With $\mathbf{x}_g^*$ standing for the global and $\mathbf{x}_{l,i}^*$ for some local optimum, we define

$$\mathrm{glc}(f) := \frac{\mathrm{avg}(f(\mathbf{x}_{l,i}^*)) - \mathrm{avg}(f(\mathbf{x}))}{f(\mathbf{x}_g^*) - \mathrm{avg}(f(\mathbf{x}))} \quad . \tag{2}$$

It follows from the definition that a low *GLC* value means a large function value difference between the average local optimum and the global optimum and thus a simple problem, whereas a high *GLC* stands for a small difference, creating a difficult problem.

2.2 Problem Generation

For setting up benchmark problems with given number of optima, *GLC* and *BSC* measures, we employ the real-valued n-Peak problem generator described in [14]. It builds fitness landscapes by combining several polynomial peaks of different height and shape. To keep things simple, we restrict it to quadratic peaks. Each peak is randomly located within the search space and possesses a maximum height between 0 and 1 (1 for the unique global optimum). Its extension in each dimension is controlled by a randomly drawn radius between $r_{\min}$ and $r_{\max}$, and by applying a randomly drawn rotation matrix we ensure non-separability of the resulting problem. All peaks are distributed in the space bounded by $[0, 20]^{D=5}$.

Table 1: Established test problem configurations of factors *peaks*, *glc*, and *bsc*. Unnumbered combinations could not be achieved by means of the problem generator (see text).

no	peaks	glc	bsc	no	peaks	glc	bsc	no	peaks	glc	bsc
1	10	0.25	0.1		10	0.25	0.01		10	0.25	10^{-3}
2	30	0.25	0.1	11	30	0.25	0.01		30	0.25	10^{-3}
	100	0.25	0.1	12	100	0.25	0.01	21	100	0.25	10^{-3}
4	10	0.50	0.1	13	10	0.50	0.01	22	10	0.50	10^{-3}
5	30	0.50	0.1	14	30	0.50	0.01	23	30	0.50	10^{-3}
	100	0.50	0.1	15	100	0.50	0.01	24	100	0.50	10^{-3}
7	10	0.75	0.1	16	10	0.75	0.01	25	10	0.75	10^{-3}
8	30	0.75	0.1	17	30	0.75	0.01	26	30	0.75	10^{-3}
	100	0.75	0.1	18	100	0.75	0.01	27	100	0.75	10^{-3}

For 5 dimensions, we realize all achievable combinations of number of peaks $= \{10, 30, 100\}$, $GLC = \{0.25, 0.5, 0.75\}$, and $BSC = \{0.1, 0.01, 0.001\}$ by measuring the resulting *BSC* and *GLC* with large random samples ($2 \cdot 10^5$) and iteratively correcting them according to simple heuristic rules. E.g., the distance between minimum and maximum basin size as provided to the generator is enlarged if the resulting *BSC* is too large, and downsized if it gets too small. A purely random process was found to be much too slow to generate the desired instances in acceptable time.

However, 6 combinations could not be realized with good accuracy as they are somewhat too extreme. They are too fragile to be possibly reached by any stochastic generator, however they may be constructed. These go unnumbered in Table 1 and are disregarded in the following. Due to the box constraints (and the cutting of peaks at the edges), it is very difficult to achieve volume differences of only $1 : 10$, and it gets even harder the more basins exist. Therefore, combinations 10, 19, and 20 are unavailable with this generator. For such a large *BSC* value, basin sizes are all relatively similar. Demanding a large *GLC* value becomes problematic in the presence of many peaks, because peak shapes are all quadratic. This is the reason why combinations 3, 6, and 9 could not be reached.

3. BASIN IDENTIFICATION METHODS

We examine how three basin identification (clustering) algorithms under different settings perform on the generated function instances given in table 1. We consider the Jarvis-Patrick (JP) clustering method ([9]), the nearest-better (NB) approach that also includes fitness information ([15]) and a procedure that goes back to the method of [20] and also refers to the topological information in order to detect multimodality. Note that whether the latter two may be applied on an initial random sample, the Jarvis-Patrick clustering as any other search space based clustering needs a developed sample that already contains a clustered structure because it does not take fitness values into account. Neither of the three methods assumes that the number of clusters/basins of attraction to find is known but detects it from the data. Some more nearest-neighbor based clustering methods have been applied inside EAs, e.g. [18], but are disregarded here. Most other methods rely on fixed size or adapted radii for separating populations but cannot provide explicite basin information and are unable to learn the number of basins actually present, e.g. [16].

The JP algorithm is a non-iterative type of clustering, based on a "nearest neighbor" mechanism. J nearest neighbors, in terms of (Euclidean) distance, are collected for each individual. Every two pairs of samples are taken and, if the two are contained in each other's neighbor list and have at least K neighbors in common, they are placed in the same cluster. The NB clustering mechanism also relies on a non-iterative procedure and a "nearest neighbor" principle, however topological information is included in addition to location of points. For each individual, it considers the connection to the nearest neighbor which is strictly better in terms of fitness. NB essentially assumes that the best individuals in different attraction basins are much further away from each other than the average distance of all individuals to their nearest better neighbors. The connections longer than $\phi \cdot mean$(lengths of all edges) are removed and the prototypes for each cluster are represented by those individuals that do not connect to others. The parameter ϕ is usually set to 2 as a robust default.

The third approach is detailed in [17], we only give an overview here. The confirmation that two points follow the same (maximum) peak follows a mechanism based on the topological positioning of the individuals in question, called detect-multimodal (DM) algorithm. We modify the *hill-valley* procedure originally proposed by [20] that chooses a set of interior points between the two on the base of user-defined gradations. A first point is set in the middle of the two, then, up to a maximum predefined number of points, another one is chosen in the middle of two gathered consecutive points (ev-

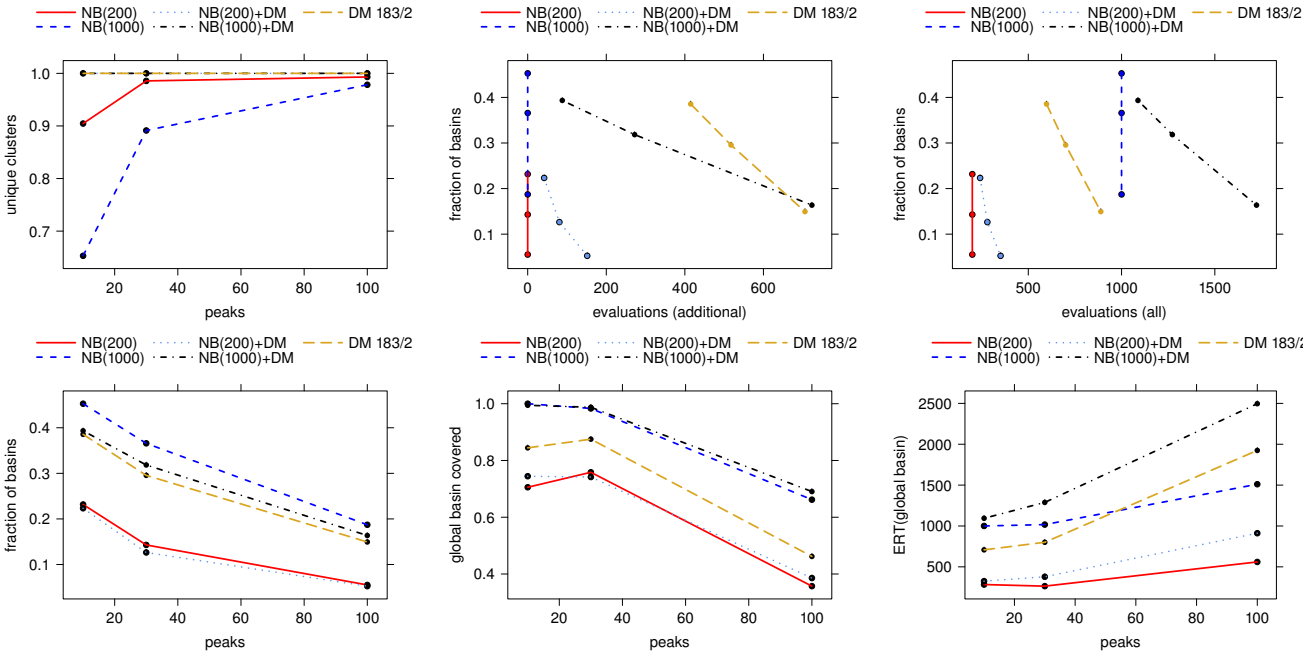

Figure 1: (Experiment 1) Comparison of the basin identification methods on the 21 test problem instances. In all figures, each method corresponds to three points, one for each basin number of 10, 30, and 100. Upper line figures from left to right: Ability to find unique clusters (only one cluster per basin), fraction of basins recognized, only considering additional evaluations (initiated by the basin identification method), fraction of basins, all evaluations (also considering the initial sample). Bottom line figures: fraction of basins recognized over the number of peaks, probability of covering the basin containing the global optimum, estimated expected runtime for detecting the basin of the global optimum.

ery so far formed disjunctive interval) with the largest gradient between them. By largest gradient we mean the approximation of the minimum ratio of the fitness distance to the (Euclidean) distance for every connection of two successive already established points. If the fitness values of all interior points are higher than the minimal fitness of the two individuals, it is concluded that they track the same optimum, otherwise they follow different peaks. When clustering is applied, the solutions are ordered in decreasing order with respect to the fitness. Then the first solution is considered the main representative of the first cluster, it is marked as processed, and is tested with DM whether it follows the same peak with any other unmarked individual, in turn. When such solutions are found, they are also marked as processed and are assigned to the same cluster. Then, the next unmarked individual is taken and the process resumes. Note that DM needs additional evaluations, whilst the first two methods do not.

For both NB and JP we also try variants in which DM is applied to the finally found clusters with the purpose of removing the overestimation of the number of existing basins. These variants are referred next as NB+DM and JP+DM, respectively. Thus, the cases of attraction basins with two or more clusters within are tackled. The number of interior points in this case is set to only 2, so that the spent number of fitness evaluations is not high, while still a reasonably accurate result is returned by DM.

4. EXPERIMENTAL COMPARISON ON THE ESTABLISHED PROBLEMS

The different basin identification methods are tested on the set of generated problems under two conditions: On random samples as they are usually employed for starting EAs, and on the best samples of a larger subset, thereby simulating a started optimization process. Note that all implementations of the chosen methods were performed by the authors. In the following, we refer to the fraction of detected basins as **FB**, and to the probability to detect the basin of the global optimum as **PG**.

Experiment 1: Which method is suited best for initial random samples? And is there a consistent effect of bsc and glc settings?

Pre-experimental planning. After trying out different parameter configurations for the DM mechanism, we decided to keep only one of 30 Latin Hypercube Sampling (LHS)-generated settings (for LHS see e.g. [11]), namely the one with an initial sample size of 183 and 2 intermediate points, as its sample size is comparable to a NB start population of 200 and the low number of intermediate points is a good compromise between quality and the needed additional evaluations. Whenever DM is applied on the result of another basin identification method, we keep using 2 intermediate points.

Task. To clearly decide for one method being better for one target value (FB or PG measures), we require that its value is consistently higher over all tested peak numbers. Concerning the effect of BSC and GLC settings, we only consider behaviors consistently found over all tested methods.

Setup. We run 5 basin identification method variants on the 21 problem instances given in Table 1, namely DM-183/2, NB(200), NB(200)+DM, NB(1000) and NB(1000)+DM. These get randomly initialized start populations of size 183, 200, and 1000, respectively. JP cannot be run on random samples. In addition to the two target values given above, we record the number of total evaluations and the number of evaluations beyond the initial sample, and also compute a very rough *expected runtime* (ERT) estimation for detecting the basin of the global optimum by simply dividing the

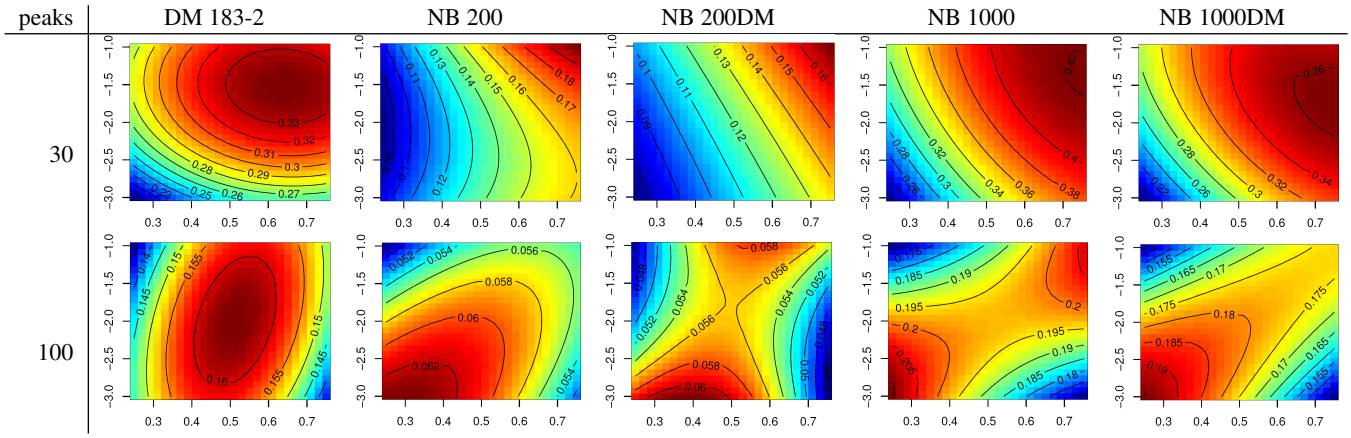

Figure 2: (Experiment 1) (Quadratic) Response Surface Model of FB measure (fraction of recognized basins), for the functions with 30 peaks (first line, 8 *glc/bsc* combinations) and 100 peaks (second line, 7 *glc/bsc* combinations), based on factors *GLC* (on the horizontal) and *BSC* (vertical). *Bsc* values scaled logarithmically.

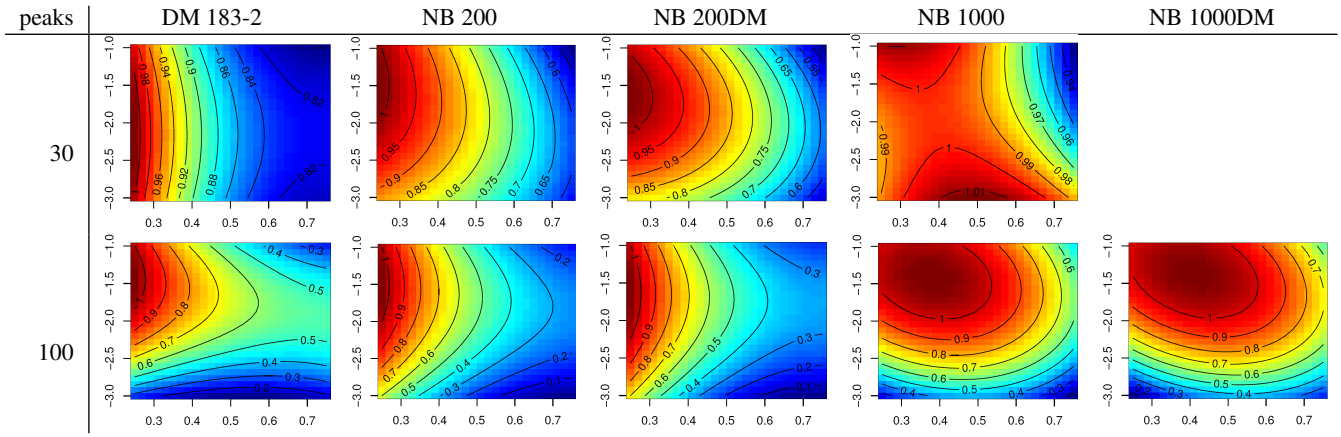

Figure 3: (Experiment 1) PG measure, modeled by a RSM for the functions with 30 (top) and 100 (bottom) peaks, based on factors *GLC* (horizontal) and *BSC* (vertical). In the missing plot for NB 1000DM on 30 peaks, values would be flat at 1. *Bsc* values scaled logarithmically.

total number of used evaluations by the probability for detecting the latter in one run.

Results/Visualization. The obtained performance data is visualized in Fig. 1 from several perspectives. Note that in the plots over the number of evaluations, the three data points for each method refer to the aggregated data on 10, 30, and 100 peaks, respectively, where the order is canonical (100 peaks always using the same or more evaluations than 30 and so on). Note that we separately plot the number of additional evaluations as these are the one that have to be invested on behalf of the basin identification method whether the initial ones are considered free as they consist of the initial population that must be evaluated anyway. To better understand the effect of *BSC* and *GLC* settings, *response surface models* (RSM) are built separately for each method variant and for the 30 and 100 peaks data. For 10 peaks there are not enough data points to establish a quadratic model, and we presume that a linear model would be unappropriate. RSM contour plots are given in Fig. 2 for the FB, and in Fig. 3 for the PG measure. However, these models are not very accurate and shall only be interpreted as trends.

Observations. In Fig. 1, we can observe that DM or any combination with DM leads to perfectly separated, unique clusters, but sometimes basins get lost during the DM cluster merge process. Generally, DM performs much better than NB on the same sample size, but also comes at a higher cost, especially for high numbers of basins. Interestingly, attaching DM to NB leads only to a small additional number of evaluations. The probabilities for detecting the basin of the global optimum are remarkably high. From the ERT plot we can estimate that probability 1 will be approached for around NB(500) even for the 100 peaks problem. But, for growing sample sizes put into NB, the fraction of unique clusters also drops, which can be cured by attaching DM (at the cost of losing some basins). The RSM plots for FB lead to the conjecture that the peak number has a very strong influence here, so 30 and 100 peaks lead to different relations. For 30 peaks, the setting with lowest *GLC* and *BSC* $(0.25,10^{-3})$ is most difficult, whether for 100 peaks, $(0.25,10^{-1})$ and $(0.75,10^{-3})$ are the hardest. For the PG measure, the dependencies seem to be much simpler. Generally, large *GLC* values lead to more difficult problems, and for 100 peaks we can also observe that smaller *BSC* values only add to this trend. However, all differences are much smaller for high peak numbers.

Discussion. Concerning the tested method variants, we may conclude that which method is appropriate depends on the allowed

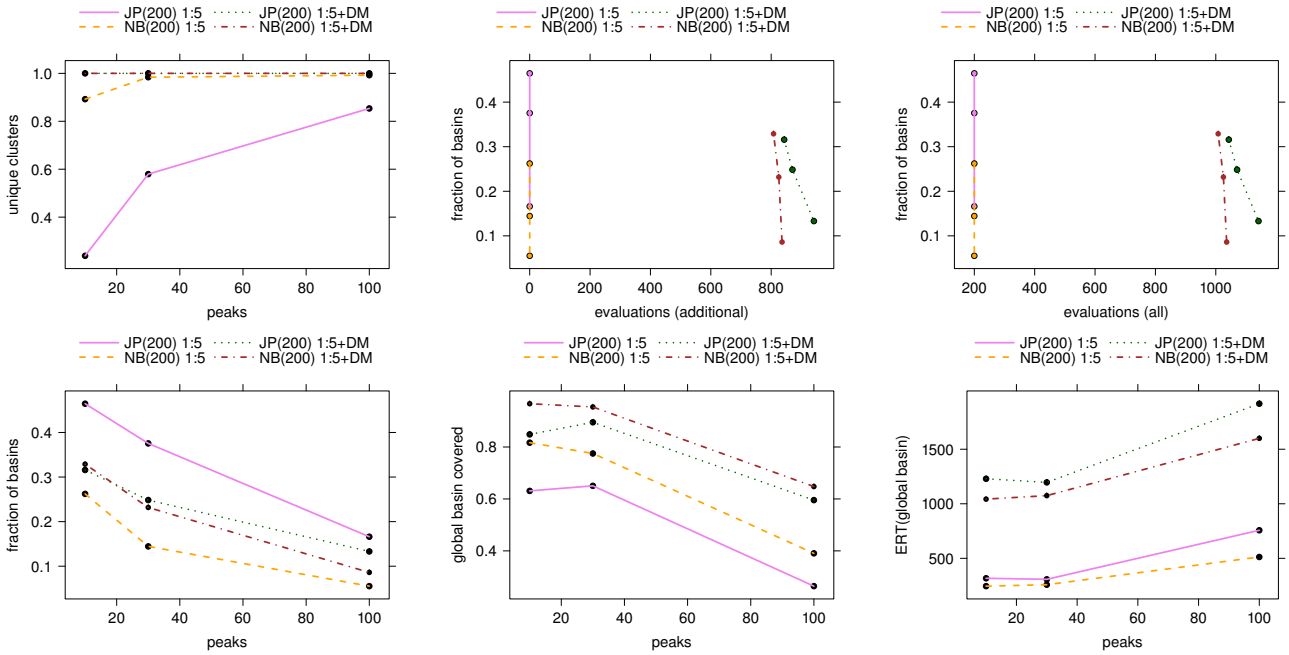

Figure 4: (Experiment 2) Comparison of the basin identification methods on the 21 test problem instances, using developed samples (best 200 of 1000). In all figures, each method corresponds to three points, one for each basin number of 10, 30, and 100. Upper line figures from left to right: Ability to find unique clusters (only one cluster per basin), fraction of basins recognized, only considering additional evaluations (initiated by the basin identification method), fraction of basins, all evaluations (initial and initiated). Bottom line figures: fraction of basins recognized over the number of peaks, probability of covering the basin containing the global optimum, estimated expected runtime for detecting the basin of the global optimum.

number of evaluations. For higher budgets, DM is the method of choice, but if we want to attain a clustering without doing additional evaluations, NB is a good alternative. Adding DM to NB results has only moderate cost and should be considered to get a more accurate cluster to basin matching, otherwise one may obtain several clusters in one basin. If the approximate number of basins is known beforehand, an appropriate sample number may be guessed from our data, but this will usually not be the case for a real-world problem. Concerning the problem properties, we can first state that both *BSC* and *GLC* obviously have some influence on problem hardness, and that the effect of *GLC* is strongest on the ability of the methods to reach good values for PG, while *BSC* has more effect on FB. However, the peak number has a very strong influence and may even change the effects of *BSC* and *GLC*.

Experiment 2: Which method is suited best for developed samples? Are there differences concerning the bsc and glc effects?

Pre-experimental planning. As we wanted to be able build RSM models for the 10 peaks settings as well, we added 4 'middle' configurations, namely all combinations of *BSC* values 0.03 and 0.003 ($\log 10 \approx -1.5$ and -2.5) and *GLC* values 0.375 and 0.625. Note that these are only used in the model, not in the comparison plots of figure 4.

Task. Similar to experiment 1, we only look for consistent similarities over all methods regarding performance and effects of bsc and glc. Concerning these effects, we also check for consistency between experiment 1 and the new data.

Setup. To remove the need to tune the JP method that possesses 2 parameters, a space-filling sample of the algorithms variables is generated via LHS. Variables J and K are allowed between $\{1,...,25\}$, but with the constraint that $J > K$. We consider 30 configurations and work with the obtained average performance. An in-run situation is simulated by generating a random sample of 10^3 individuals and selecting the best 200 from them (1:5). On this set, we run 4 method variants (again on the 21 problem instances of Table 1), namely JP(200), JP(200)+DM, NB(200), and NB(200)+DM. Data is recorded in the same way as for Experiment 1.

Results/Visualization. Fig. 4 shows the performance comparison of the methods, Figs. 5 and 6 the RSM plots over measures FB and PG.

Observations. The performance plots offer some interesting findings: The JP method achieves the best FB results, but with many non unique clusters, which is undesired. The amount of evaluations spent for additional DM runs is much smaller and more stable than in Experiment 1. JP and NB both profit a lot from applying DM when the PG measure is considered, but even without, the values obtained by NB are already good (despite JP finding many more basins on average). In the RSM plots, we find that the FB hint to a different behavior of JP and NB: Where we can recognize a high correlation between *BSC* and quality for JP, NB seems to depend more on *GLC*. For 10 peaks, this influence is not very strong, but for 30 and 100 peaks, it clearly shows. Concerning problem hardness, we make a slightly different observation compared to experiment 1. Here, the hardest problem seems to be $(0.75,10^{-3})$ only, where it depends on the basin identification method how difficult the other extreme points are. For the PG measure, the generated RSM look very similar for 30 and 100 peaks and not much different from the plots obtained in Experiment 1. Problems tend to be harder for large gsc and/or small bsc values. However, for 10 peaks, we have a different picture, and all 4 methods seem to behave very similar.

Discussion. The comparison of JP and NB shows they have differ-

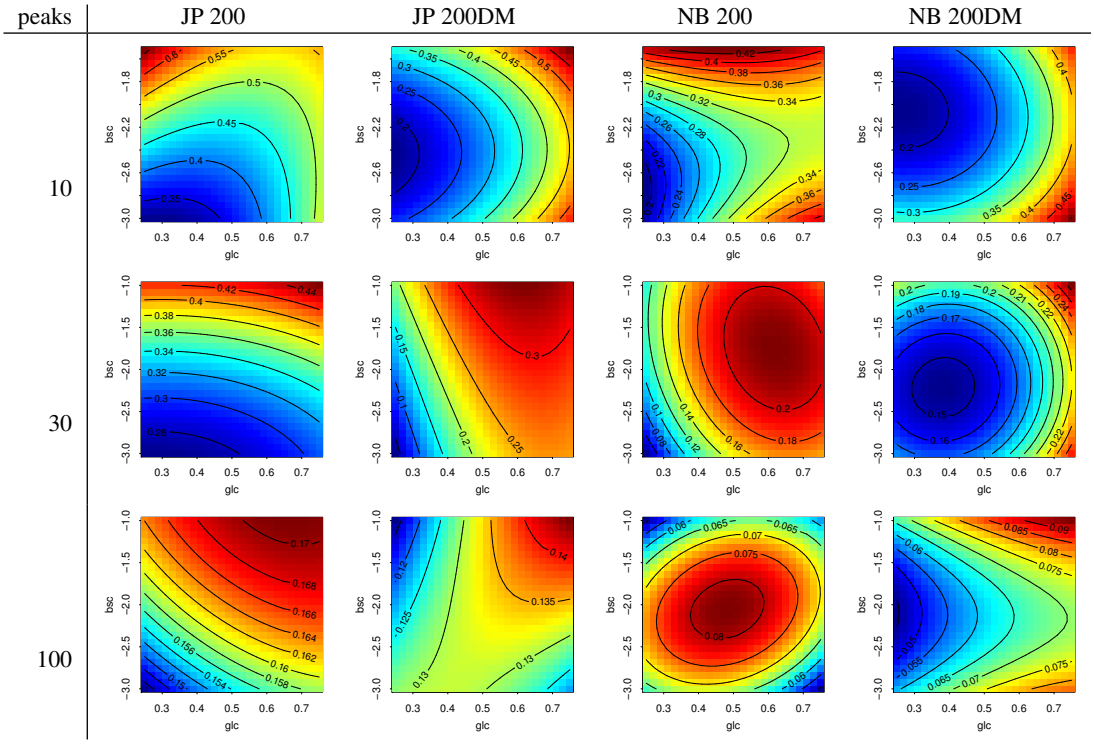

Figure 5: (Experiment 2) FB measure, modeled by a RSM for the functions with 10, 30 and 100 peaks, based on factors *GLC* (horizontal) and *BSC* (vertical). *Bsc* values scaled logarithmically.

ent advantages: JP finds more basins, but the basin-cluster mapping is more distorted, which is probably unsuited for using it inside a niching algorithm. Applying DM leads to very similar FB and PG values for JP and NB at almost the same cost, but the needed budget is considerable (roughly 4 times the size of the initial sample). If evaluations are precious, we would therefore recommend using NB if the main task is to locate the global optimum, and JP if many good solutions are sought. For larger budgets, NB+DM or JP+DM are both acceptable. We can state that concerning bsc and glc effects, the similarities between the results obtained for different methods and base data (in Experiment 1) are relatively strong and thus conclude that both effect hardness of a problem. Nevertheless, the peak number has the strongest effect and especially for low peak numbers completely dominates the quality of the basin identification methods, regardless of the other factors. For larger peak numbers, this changes and *BSC*, *GLC* and the method itself gain weight. However, while this may be surprising, it can be only considered as trend and shall be investigated further.

5. CONCLUSIONS AND OUTLOOK

Following from the discussions of our experimental results, we can conclude that the two properties used to establish test problem instances indeed have some effect on problem hardness, at least insofar as basin identification methods are considered. The large *gsc* and small *BSC* values affect the ability of these methods to detect the basin that holds the global optimum, while the effect on the number of detected basins appears to be more complex. In either case, the peak number strongly interacts with both properties' effects, and for low peak numbers, completely hides them. For higher peak numbers (in our investigation $\geq$ 30), the effects of the other two properties get stronger. This interaction shall open up an interesting research line for future investigations. Concerning the basin identification methods, we can make recommendations based on the available budget. If we have an initial sample and evaluations are precious, NB seems to be the method of choice. If we can invest a bit more, DM alone, NB+DM or JP+DM are reasonable alternatives. For a developed sample (within a run), JP is remarkably strong, but concentrating more on finding many good peaks than on the globally optimal one, whether NB does it the other way around. However, all these methods are simple heuristics and it would be rewarding to attempt improving them in the future, possibly by recombining them.

References

[1] B. Addis. *Global Optimization using Local Searches*. PhD thesis, Department Systems and Computer Science, of University of Florence, 2005.

[2] A. Auger, S. Finck, N. Hansen, and R. Ros. BBOB 2010: Comparison Tables of All Algorithms on All Noiseless Functions. Technical Report RT-388, INRIA, 09 2010.

[3] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005)*, pages 1769–1776. IEEE Press, 2005.

[4] K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When is "nearest neighbor" meaningful? In *International Conference on Database Theory (ICDT 1999)*, pages 217–235, London, UK, 1999. Springer-Verlag.

[5] M. Giacobini, M. Preuss, and M. Tomassini. Effects of scale-free and small-world topologies on binary coded self-adaptive CEA. In J. Gottlieb and G. R. Raidl, editors, *Sixth European Conf. Evolutionary Computation in Combinatorial Optimization, Proc. (EvoCOP'06)*,

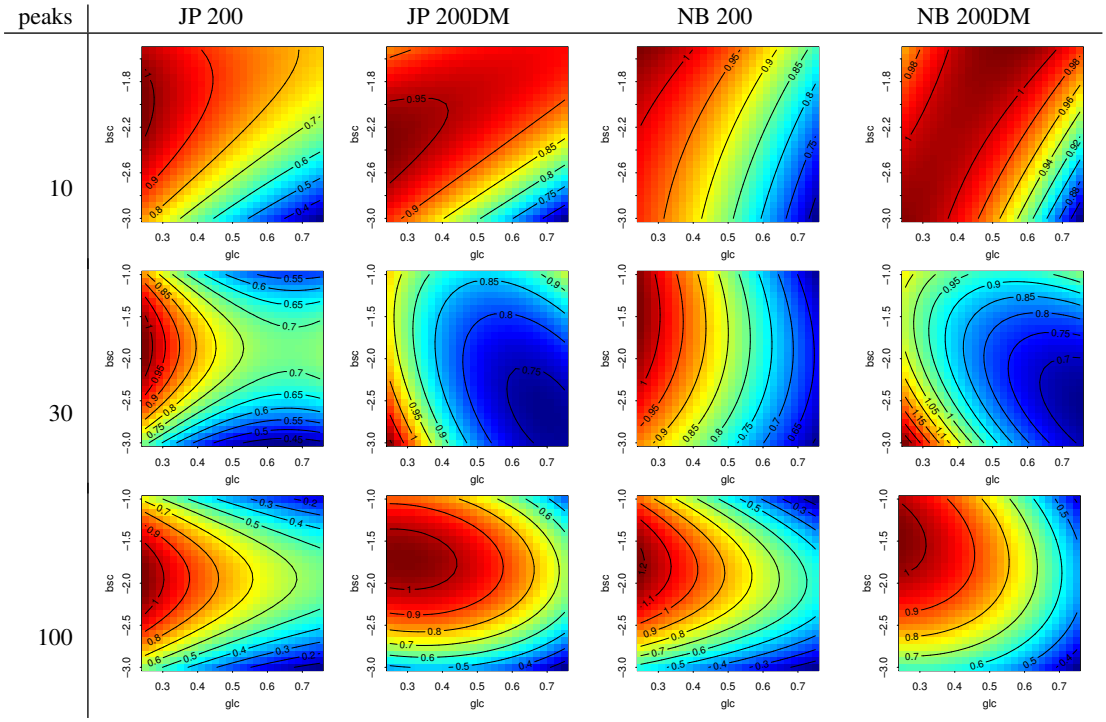

Figure 6: (Experiment 2) PG measure modeled by a RSM for 10, 30 and 100 peaks, based on factors *GLC* (horizontal) and *BSC* (vertical). *Bsc* values are scaled logarithmically.

volume 3906 of *Lecture Notes in Computer Science*, pages 86–98, Berlin, 2006. Springer.

[6] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In *Workshop Proceedings of the GECCO Genetic and Evolutionary Computation Conference*, pages 2389–2395. ACM, July 2009.

[7] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009.

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

[9] R. Jarvis and E. Patrick. Clustering using a similarity measure based on shared near neighbours. *IEEE Transactions on Computers*, 22(11):1025–1034, 1973.

[10] O. Mersmann, M. Preuss, and H. Trautmann. Benchmarking evolutionary algorithms: Towards exploratory landscape analysis. In R. Schaefer et al., editors, *PPSN XI: Proceedings of the 11th International Conference on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science 6238, pages 71–80. Springer, 2010.

[11] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, 5th edition, 2001.

[12] M. Preuss. Niching Prospects. In *Proceedings of Bioinspired Optimization Methods and their Applications (BIOMA 2006)*, pages 25–34. Jozef Stefan Institute, Ljubljana, Slovenia, 2006.

[13] M. Preuss and T. Bartz-Beielstein. Experimental analysis of optimization algorithms: Tuning and beyond. In Y. Borenstein and A. Moraglio, editors, *Theory and Principled Methods for Designing Metaheuristics*. Springer, 2011.

[14] M. Preuss and C. Lasarczyk. On the importance of information speed in structured populations. In X. Yao et al., editors, *Parallel Problem Solving from Nature - PPSN VIII*, volume 3242 of *LNCS*, pages 90–99, Berlin, 2004. Springer.

[15] M. Preuss, L. Schönemann, and M. Emmerich. Counteracting genetic drift and disruptive recombination in $(\mu + /, \lambda)$-EA on multimodal fitness landscapes. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005)*, volume 1, pages 865–872, New York, 2005. ACM Press.

[16] O. M. Shir and T. Baeck. Niche radius adaptation in the cma-es niching algorithm. In *Proceedings of Parallel Problem Solving from Nature (PPSN 2006), Lecture Notes in Computer Science*, volume 4193, pages 142–151, Berlin, Germany, 2006. Springer-Verlag.

[17] C. Stoean, M. Preuss, R. Stoean, and D. Dumitrescu. Multimodal optimization by means of a topological species conservation algorithm. *IEEE Transactions on Evolutionary Computation*, 2010. in press.

[18] F. Streichert, G. Stein, H. Ulmer, and A. Zell. A clustering based niching ea for multimodal search spaces. In P. Liardet, P. Collet, C. Fonlupt, E. Lutton, and M. Schoenauer, editors, *Artificial Evolution, 6th International Conference, Evolution Artificielle, EA 2003, Marseilles, France, October 27-30, 2003*, volume 2936 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2003.

[19] A. Törn, M. Ali, and S. Viitanen. Stochastic Global Optimization: Problem Classes and Solution Techniques. *Journal of Global Optimization*, 14(4):437–447, 1999.

[20] R. K. Ursem. Multinational evolutionary algorithms. In *Proceedings of Congress of Evolutionary Computation (CEC 1999)*, volume 3, pages 1633–1640, Piscataway, NJ, 1999. IEEE Press.

[21] M. Wineberg. *Improving the Behavior of the Genetic Algorithm in a Dynamic Environment*. PhD thesis, Carleton University, Ottawa, Canada, 2000.