Towards Scalability in Niching Methods

Marcel Kronfeld and Andreas Zell

***Abstract*— The scaling properties of multimodal optimization methods have seldom been studied, and existing studies often concentrated on the idea that all local optima of a multimodal function can be found and their number can be estimated *a priori*. We argue that this approach is impractical for complex, high-dimensional target functions, and we formulate alternative criteria for scalable multimodal optimization methods. We suggest that a scalable niching method should return the more local optima the longer it is run, without relying on a fixed number of expected optima. This can be fulfilled by sequential and semi-sequential niching methods, several of which are presented and analyzed in that respect. Results show that, while sequential local search is very successful on simpler functions, a clustering-based particle swarm approach is most successful on multi-funnel functions, offering scalability even under deceptive multimodality, and denoting it a starting point towards effective scalable niching.**

I. Introduction

Metaheuristic optimization methods are a class of general search mechanisms applicable to almost any problem instance which can be expressed through computational means. An optimization problem is formulated on a target function $f : D \to M$ for an ordering relation $\leq$ as the problem of finding the minimum $x_0 \in D$ of f: $\forall x \in D : f(x_0) \leq f(x)$. In the more specific area of single-objective, real-valued function optimization, we assume that f operates on $\mathbb{R}$, $f : \mathbb{R}^n \to \mathbb{R}$. Metaheuristic approaches are especially useful when facing non-linear, non-convex or non-continuous optimization problems, which are difficult to handle using classical optimization. They are often inspired by natural processes such as the Darwinian theory of evolution. They can also be extended to work robustly on noisy or time-dependent target functions.

A frequent challenge for heuristic optimization is the non-convexity of the target function, which implies the existence of local optima. A local optimum can be defined using a notion of neighborhood within the domain of f. x_0 is defined as a local minimum of f if $\forall x \in U(x_0) : f(x_0) \leq f(x)$ for an open neighborhood set $U(x_0)$. However, in the continuous domain, the number of neighbors is usually infinite, so that local optimality in general can only be assessed with some probability. Hence often a degree of local smoothness of f is assumed, which can be exploited by a local search method to converge on a local optimum if initialized in its vicinity.

Assuming that local optimality can be assessed, most global direct search methods concentrate on finding a single best solution, hoping it to be the global optimum of f. Quite often, however, practitioners are interested in a larger set of relatively good solutions, or even all local solutions if their number is small enough. Examples are applications from physics [1] or bioinformatics [2]. In Metaheuristic Multimodal Optimization (MMO), this task is tackled by either iterative or parallel search for multiple local optima. From the context of evolutionary inspired metaheuristics, MMO is often related to the notion of *niching*. An ideal niche can be defined in relation to a local search operator m as the largest neighborhood within which x_0 is locally optimal, $\hat{U}_{\mathrm{m}}(x_0) = \{x \in D | \exists k \in \mathbb{N} \forall j \geq k : \mathrm{m}^j(x) = x_0\}$, meaning that any local search initialized within that niche (or neighborhood) will converge on the same center of attraction, x_0. Therefore, $\hat{U}(x_0)$ is also called a *basin of attraction*. A *niching method* in MMO now seeks to estimate or identify several such basins of attraction, which is a prerequisite to identify several local optima.

Iterative global search, also called *sequential* or *temporal niching*, since niches are identified sequentially in time, mainly bares the risk of finding the same local optimum several times, thus wasting valuable optimization time. In addition to that, a global search run is, practically by definition, expected to visit several basins of attraction during a run. Since only the single best solution is returned in the end, all but one of them will be discarded. This indicates that a niching variant of a global search metaheuristic will most probably be more efficient in finding multiple optima than the simple iterated approach in terms of evaluations required.

While current MMO approaches usually require a good estimate on the number of true local optima, we presume this to be difficult for unknown target functions. In practice, the number of local optima can be very high, making it infeasible to find all of them. MMO methods should instead provide the best-so-far local optima even if only a short optimization time can be afforded. Thus, this work approaches the idea of *anytime-criteria* in MMO, without the necessity of an *a priori* estimate on the number of local optima.

A. Related Work

Multimodal optimization has been a front topic of research in metaheuristic optimization for some time. Initially, techniques for finding multiple optima in parallel were concerned with keeping up diversity in Genetic Algorithms (GAs). GAs are a typical metaheuristic optimization method, working on binary strings that represent a set of candidate solutions. This population is iteratively improved taking into account the quality (fitness) of the candidates. The heuristic GA operators are in analogy to Darwinian evolution: fitness-based selection, genetic mutation and recombination. Evolution Strategies (ES) and Differential Evolution (DE) are similarly based on evolutionary schemes. Others, such as Particle Swarm Optimization

Marcel Kronfeld and Andreas Zell are with the Wilhelm-Schickard-Institute for Computer Science, University of Tübingen, Sand 1, 72076 Tübingen, Germany, {marcel.kronfeld,andreas.zell}@uni-tuebingen.de

(PSO), simulate swarming behavior in animals and variate candidate solutions by attraction towards better neighbors.

All of these methods iteratively improve the population hoping to converge on a solution which is at least locally optimal. In taking several candidates in parallel into account, metaheuristics have a higher chance of the found solution being globally optimal when compared to local search methods. A detailed introduction to metaheuristics is presented in [3].

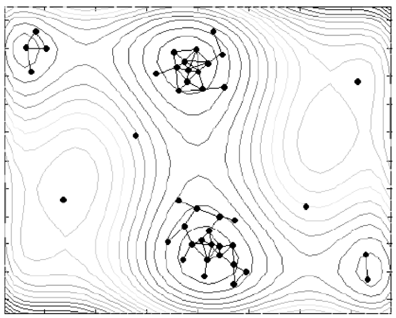


Fig. 1. Visualization of a niching approach, from [4].

Niching in GA probably started with the crowding technique [5], which requires a new candidate solution to replace a very similar, worse solution in the population to maintain diversity. The sharing approach [6] reduces the selection probability of similar individuals by taking the number of close-by solutions into account. The more individuals are gathered within a distance of σ, the worse is their chance for survival. The distance threshold σ can be interpreted as an early appearance of a niche definition: individuals within a σ-environment are assumed to occupy one niche, and by limiting the number of individuals per niche, several niches can implicitly be covered.

Next to several extensions to sharing and crowding, such as iterative clearing [7], deterministic crowding [8], and restricted tournament selection [9], iterative niching with modification of the search space has also been presented [10], [11]. Techniques of explicit parallel niching by building sub-populations or dividing the search space were brought up in several variants [12], [13] and have been introduced into other branches of metaheuristic optimization, such as ES [14], DE [15], and PSO [16], [17]. A general approach has been presented in [4], which allows the use of generic optimizers on sub-populations formed by clustering (Fig. 1).

II. Scalability Criteria for MMO

This work aims at a discussion of the behavior of current niching methods in higher dimensions and an evaluation on some exemplary benchmarks. Parallel multimodal optimization methods are often designed and tested on low dimensional functions in the expectation of finding all local optima of the target function. Upscaling to higher dimensions is tackled seldom, and if so, holding up this expectation. Exemplary is the work by Brits et al. [18], [19], where dimensions of two benchmark functions are increased from $n = 1$ to $n = 4$, and the population size is increased exponentially in n with the number of local optima. Population boosting was done similarly in [20]. Other approaches, such as that by Shir and Bäck in [14], have been scaled up to $n = 40$ dimensions assuming that a limited number of optima is to be found, e.g., the best $q = 81$ on a 40-dimensional function.

Both approaches have worked out well in the tested scenarios. However, we suggest that, in practice, (i) the number of interesting optima can hardly be easily estimated, and (ii) it is infeasible to search for all the local optima of a complex target function. We therefore formulate two criteria we would pose on an MMO method concerning scalability:

1) A scalable MMO method should work sequentially in the sense that, for a highly multimodal target function, increasing the iterations T of the MMO method increases the number k of local optima returned. If the number of local optima is large in relation to k, k should behave approximately linearly to T: $\frac{T_1}{T_2} \approx \frac{k_1}{k_2}$.
2) For a scalable MMO method, the minimal number of iterations required to identify a first subset of local optima should be small, or be adjustable to be small.

Both points are related to each other, bearing the consequence that the population size cannot be increased with the (often exponentially) increasing number of local optima, because this would increase the runtime until any local optimum can be found to the same extent, which is impractical. The first point aims at loosening the assumption that the number of solutions needs to be predefined. It requires an MMO method to work at least *semi-sequential*, meaning that optimum identification and exploration phases occur in parallel or are triggered within a single optimization run by the algorithm itself. These properties can also be interpreted as *anytime* requirements towards the MMO concerning monotonicity and interruptibility: A scalable niching method should provide a reasonable set of solutions at any time it is halted by the user, who expects to receive more results after longer runtimes.

Sequential niching methods realize these criteria in a straight-forward way, since they iterate global search and convergence phases ad infinitum. For this class of MMO methods, collision avoidance mechanisms that avoid finding the same optimum again and again have been considered [10], [11]. However, for high dimensions, their merit is questionable unless there are predominant niches, in which the optimizer gets trapped repeatedly. For highly multimodal target functions without such predominant basins, the probability of finding the same local optimum several times is very low for typical runtimes. Thus, although collision avoidance methods are appropriate in some circumstances, they are not regarded here.

For parallel niching methods, the scalability criteria can be fulfilled by introducing automatic restarts. As soon as a local optimum has been identified, it is stored and the corresponding search capacity is reused. This has also been called *niche deactivation* in [4], [2], where converged individuals are reinitialized across the search space and resume exploration.

A. Algorithms under Study

In the following, we will present several MMO methods and analyze and compare their abilities to fulfill the scalability criteria. Among basic sequential methods, we chose to compare Iterative Nelder-Mead-Simplex (INMS) and IPOP-CMA-ES. INMS is simply the iteration of a Nelder-Mead-Simplex search [21] which is reinitialized as soon as the current solution has converged for a fixed number of evaluations. IPOP-CMA-ES [22] is a very successful ES with covariance matrix adaption with restarts. If the ES converges during a run, it is automatically reinitialized, at the same time increasing the population size by a given factor c. We try both a moderate and the standard increase factor ($c \in \{1.2, 2\}$).

Among specialized MMO algorithms, we tested two niching swarm methods, NichePSO [16] and ANPSO [17]. NichePSO treats single PSO particles as local searchers. As soon as a singular particle has converged, a subswarm is formed with its closest neighbor. The subswarms are expected to merge in more promising areas, but such events will be infrequent with limited population sizes (cf. criterion 2). As NichePSO also lacks a global exploration component, we expect it to be inferior in higher dimensions. ANPSO, an adaptive extension of NichePSO, reintroduces a main swarm for global exploration and forms subswarms based on an adaptive distance parameter computed from the population diversity. This allows both for larger subswarms and a better explorative behavior. Both techniques were extended by subswarm-deactivation to being able to conform to the scaling MMO criterion 1. Therefore, population sizes were selected relatively small ($\lambda = 50$).

As a second type of MMO algorithms, we employed the generic Clustering-based EA (CBN-EA) [4]. The CBN-EA uses a clustering method on the current population to identify niches, and optimizes each sub-population with an instantiation of a generic metaheuristic. Those individuals which cannot be assigned to a cluster make up the main population which explores the search space. CBN-EA also reinitializes converged sub-populations to the main population. Combining density-based clustering with DE [23], PSO [24], GA [25] and CMA-ES [26], four variants of the CBN-EA were tested. Since most metaheuristics are infeasible to be run with small populations, which can occur due to the clustering in a CBN run, we limited the sizes of sub-populations within $[10, 15]$. If a cluster S_i grows larger, namely $s_i = |S_i| > 15$, the $s_i - 15$ worst individuals are reinitialized to the main population. The clustering method from [4] was applied with a density parameter of $\sigma = 0.1$ relative to the problem range, which implies that local optima lying closer than $\sigma_r = 10^{-1}(r_u - r_l)$ for the problem domain $[r_l, r_u]^n$ can not be distinguished by the CBN-EA. This resolution seems coarse, yet due to the curse of dimensionality, the hyper-cube of side-length σ_r covers only a 10^{-n}-th of an n-dimensional search space, making the resolution sufficiently fine-grained in higher dimensions. Also note that, due to the PSO variation mechanism working based on attraction towards earlier positions, the CBN-PSO variant employed the clustering on those memorized positions per individual, and not on the current particle positions. Otherwise, the CBN-EA variants work analogously. The following list subsumes the employed MMO algorithm configurations. All methods employed a population size of $\lambda = 50$, except for the IPOP-ES which started with $\lambda = 4 + \lfloor 3\ln(n) \rfloor$, $\mu = \left\lfloor \frac{\lambda}{2} \right\rfloor$.

1) NPSO: NichePSO with $\phi_1 = 1.2, \phi_2 = 0, \omega(t) = 0.7 - \frac{t}{t_{max}} 0.5$, fully connected constricted GCPSO ($\phi_1 = \phi_2 = 2.05$, $\chi = 0.73$, $\rho = 0.1$) for subswarms [16].
2) ANPSO: NichePSO with adaptive niche radius and constricted PSO ($\phi_1 = \phi_2 = 2.05$, $\chi = 0.73$) for the main swarm (grid neighborhood) [17].
3) 1.2-IPOP: CMA-ES with increasing population size (increase factor $c = 1.2$) [22].
4) 2-IPOP: IPOP-CMA-ES with $c = 2$.
5) CBN-PSO: clustering-based niching PSO ($\phi_1 = \phi_2 = 2.05$, $\chi = 0.73$, grid neighborhood).
6) CBN-DE: clustering-based niching DE/current-to-best/2 ($F = 0.8$, $k = \lambda_{DE} = 0.6$).
7) CBN-ES: clustering-based niching CMA-ES with $\frac{\mu}{\lambda} = \frac{3}{10}$, $p_{mut} = 1$, no crossover.
8) CBN-GA: real-valued, elitist GA with tournament-of-four selection, uniform self-adaptive mutation ($p_{mut} = 1$), 1-point crossover ($p_{co} = 0.5$).
9) INMS: Iterative Nelder-Mead-Simplex, automatic restart if NMS stagnates for 15 iterations ($15 \cdot \lambda$ evaluations).

B. Performance Measurement

As to measuring performance in MMO, the Maximum Peak Ratio (MPR) measure is widely used. It assumes knowledge of all local optima $\hat{X} = \{\hat{x}_i\}_{1 \leq i \leq q}$ of f, and is defined for maximization problems on a set of candidate solutions P:

$$\text{MPR}(P, \hat{X}) = \frac{\sum_{(x_j, \hat{x}_i) \in assoc(P, \hat{X})} f(x_j)}{\sum_{i=1}^{q} f(\hat{x}_i)} \qquad (1)$$

The set $assoc(P, \hat{X}) \subset P \times \hat{X}$ consists of associated pairs of candidate solutions with local optima $\hat{x}_i$, which are formed by selecting the closest candidate from P for each $\hat{x}_i$. Note that $|assoc(P, \hat{X})| < |\hat{X}|$ is allowed if several candidates occupy the same optimum while other optima are not covered at all. The MPR lies in $[0, 1]$ and is the closer to 1 the more accurate all optima are covered. Also, it rates better optima higher due to their higher contribution to the summed-up fitness values. Because neither the number nor the location of local optima are necessarily known for complex, high dimensional target functions, we suggest an alternative performance measure.

Instead of assuming full knowledge of local optima, we select a threshold interval $[\theta_l, \theta_u]$ for minimization, covering all function values which are regarded as *interesting*. While θ_l is a lower bound to the reachable fitness values, which can often be estimated in practice, θ_u gives an upper bound below which values are judged to be interesting results, e.g., by an expert in the application. Such an approach has been chosen in [2], for example, for an application from bioinformatics.

Given the θ-interval, Eq. 2 shows a simple way to calculate a population score within $[0, s_{max}]$, where s_{max} is unknown as long as the number of local optima is unknown.

$$\text{sc'}(P, \theta_l, \theta_u) = \sum_{\{x_i \in P | f(x_i) < \theta_u\}} \frac{\theta_u - f(x_i)}{\theta_u - \theta_l} \qquad (2)$$

TABLE I

THE APPLIED BENCHMARK FUNCTIONS.

Name	Formula	Domain	Thresh. $[\theta_l, \theta_u]$ at $n = 10$, $n = 30$	
Rastrigin's	$f_{Rs}(\vec{x}) = 10n + \sum_{i=1}^{n}(z_i^2 - 10\cos(2\pi z_i))$; $\vec{z} = (\vec{x} - \vec{o})M_{Rs}$	$[-5, 5]^n$	$[-0.5, 15.5]$	$[-0.5, 79.5]$
Schwefel's Sine-root	$f_S(\vec{x}) = 418.9829 \cdot n - \sum_{i=1}^{n}(x_i sin\sqrt{\lvert x_i \rvert})$	$[-512, 512]^n$	$[0, 800]$	$[0, 3200]$
Rana's	$f_{Rn}(\vec{x}) = \sum_{i=1}^{n-1}\left[(z_i sin(a_i)cos(b_i) + (z_{i+1} + 1)cos(a_i)sin(b_i))\right]$ $a_i = \sqrt{\lvert z_{i+1} - z_i + 1\rvert}$, $b_i = \sqrt{\lvert z_i + z_{i+1} + 1\rvert}$, $\vec{z} = \vec{x}M_{Rn}$	$[-512, 512]^n$	$[-5000, -3400]$	$[-15000, -8600]$

One disadvantage of Eq. 2 on unspecific collections P is that it is prone to be mislead by redundancy: if all candidates in P_1 are gathered closely around the same local optimum, while P_2 consists of few distinct local optima, Eq. 2 could still score P_1 much higher than P_2. This is opposed to the fact that P_2 contains more information, as P_1 is highly redundant. As an alternative variant, Eq. 3 requires a clustering and a binning step on P, sorting the solutions into k bins $B_1 \ldots B_k$ covering the interval $[\theta_l, \theta_u]$. Specifically, we employ density-based clustering with parameter σ to remove redundancy.

$$\text{sc}(P, \theta_l, \theta_u) = \sum_{B_j \in \text{Bin}_k(\text{clust}_\sigma(P), \theta_l, \theta_u)} w_j |B_j| \quad (3)$$

In Eq. 3, each bin B_j is assigned a weighting factor w_j used to weigh the quality of the optima found against each other. Typically, the number of inferior optima (near θ_u) is much larger than that of high quality (near θ_l). If this distribution is known, the weights in Eq. 3 can be adapted accordingly. For our analysis, we employ equidistant binning with $k = 16$ and set $w_j = \frac{k-j+1}{k}$ for $1 \leq j \leq k$. Thus, any solution in B_1, which is the best bin, has a value of 1, while solutions in the worst class B_k add values of $\frac{1}{k}$ to the score. In Eq. 2, candidate solutions close to θ_u are widely ignored, especially for larger ranges of $[\theta_l, \theta_u]$. Thus, although being more coarse-grained, Eq. 3 complies better with the notion that even optima near θ_u are seen as *interesting*, contributing a fixed value to the score.

While this score rates the combined quality of the found candidate solutions, their accuracy remains in question. A solution is the more accurate the closer it lies to an actual local optimum of the target function f. Accuracy is hard to evaluate if the true local optima are unknown. One possibility to assess it is the use of a post-processing step refining the candidate solutions through local search. A solution can be considered accurate with respect to a local search method and a threshold ϵ if the local search process converges within the ϵ-vicinity of the candidate solution. If an optimizer finds numerous optima which are not accurate, the score results are potentially overrated, since many of the candidate solutions may lie within the same basin of attraction, which other optimizers may have identified and assigned only one local solution.

We thus look at the accuracy of a candidate solution by refining it with NMS performing $100n$ steps for an n-dimensional problem f with different threshold values $\epsilon \in \{0.01, 0.001, 0.0001\}$. The ϵ-values are interpreted relatively to the problem range, which, for the benchmarks considered, are always of the form $R = [r_l, r_u]^n$. For the NMS refinement, a local search population of size $n + 1$ is created around the candidate solution $\vec{x}_i$ by perturbing each component of $\vec{x}_i$ by $\frac{\epsilon}{2}(r_u - r_l)$ for the threshold ϵ. If NMS fails to find a better position $\vec{x}_i'$ with $d(\vec{x}_i, \vec{x}_i') > \epsilon(r_u - r_l)$, the candidate solution $\vec{x}_i$ is regarded as a local optimum within accuracy ϵ. As the online score from Eq. 3 is based on a clustered, but unrefined population, it produces optimistic values. By comparing the last online score to a final score of refined solutions, the discrepancy of the online score of an optimizer can be judged.

C. Experiments

In order to rate niching methods concerning the scalability criteria presented in Sec. II, we conducted empirical tests on eight popular multimodal benchmark functions. Next to Rastrigin's, Schwefel's and Rana's functions (Tab. I), we considered the L-function, Griewank, Fletcher-Powell, Bohachevsky [27], and the Levy function [28]. Especially for Bohachevsky and Griewank, the interesting optima are in very close vicinity compared to the problem range, shifting the difficulty to the niche-radius problem ([17], [27]), consisting in the problem-specific discrimination of local optima, which is not part of this work. Therefore, these two functions will not be regarded. Furthermore, due to space restrictions, we concentrate on three especially challenging benchmarks (Tab. I) and relate to the results on the rest of the benchmarks in Sec. III.

The well-known Rastrigin's function f_{Rs} is a modulated hyper-parabola with a large number of local optima. It is "single-funnel", meaning that it provides a global basin of attraction of second order: of two local optima, the dominant one is always closer to the global optimum. In that sense, f_{Rs} is *non-deceptive*. Rastrigin's function is shifted in its domain and rotated according to [29]. The similarly popular sine-root function f_S by Schwefel is more difficult in that it does not provide a global basin of attraction: there are various dominant local optima spread throughout the search space, prominently at combinations of $x_i \in \{-512.0, -312.52, 420.97\}$, for which any neighboring local optimum is worse in quality. f_S is separable, but it is also *deceptive* compared to f_{Rs}.

As Rana's function f_{Rn} has its dominant optima close to the corners of the search space, which provided systematic advantages to specific optimizer implementations, it is rotated by M such that $\frac{\vec{u}}{|\vec{u}|}M = \vec{e}_1$, where $\vec{u} = (1, 2, 3, ..., n)$ and $\vec{e}_1 = (1, 0, 0, ..., 0)$. The Rana function f_{Rn} is deceptive as well, and it is also non-separable. Fig. 2 shows 2D illustrations of the benchmark functions. Tab. I also shows the domains and the threshold intervals $[\theta_l, \theta_u]$ within which candidate solutions

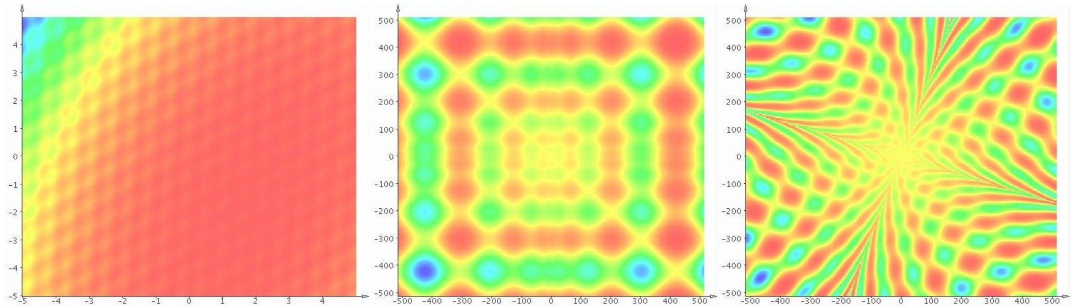


Fig. 2. Illustrations of the benchmark functions in 2D. From left to right: Rastrigin's, Schwefel's and Rana's.

are seen as *interesting*. They were selected by testing several global optimizers (DE, PSO, GA, CMA-ES, NMS) on the function and selecting the upper bound in such a way that at least two optimizers find interesting solutions consistently.

For the empirical evaluation, we performed two series of experiments setting $n \in \{10, 30\}$, with $5000n$ and $20000n$ evaluations per run. The short runs were repeated 50 times with different random seeds, the long runs 25 times.

III. Results and Discussion

Figures 3 and 4 show averaged results of the online scores achieved in the short runs. On F_{Rs}, the global basin of attraction can be exploited, visible in the success of INMS in 10-D. The global correlation is also of great help for CMA-ES, which achieves the best scores both online and after refinement in 30-D (Fig. 4, left). CBN-GA achieves the next-best results, probably due to the symmetric mutation operator employed being advantageous on F_{Rs}. However, CBN-GA lacks accuracy compared to IPOP-CMA-ES (Fig. 5 a). On the long-run experiment in 30-D, CBN-PSO is the only other niching method showing scalability on F_{Rs} (Fig. 9, left).

INMS and IPOP-CMA-ES are much less successful on F_S, where there is no global basin of attraction and the optima are near the bounds. In 30-D, they widely fail to find any solutions below the desired threshold (Fig. 4, middle); among the standard solvers, only PSO and DE reach interesting function values robustly, and hence only ANPSO and CBN-PSO/DE deliver noteworthy scores (Figs. 3-4, middle). CBN-PSO achieves best scores after final refinement (Fig. 5 b,e), where it shows that ANPSO and CBN-DE are not as accurate. In case of NPSO/ANPSO, we ascribe this to the tendency to allocate relatively few individuals per niche, which hinders local convergence. Similarly, DE typically requires larger populations to achieve close convergence. On the long run, ANPSO and CBN-PSO show best scalability properties on F_S (Fig. 10 b,e). For a notion of the distribution of local optima found on F_S-30D, Fig. 6 shows the averaged resulting histograms of the successful algorithms with increasing accuracy ($\epsilon_1 = 0.01$, $\epsilon_2 = 0.001$, $\epsilon_3 = 0.0001$, relative to $[r_l, r_u]$).

The discrepancy between online and refined score is most prominent on the rotated f_{Rn} (Figs. 3-4, right, vs. Fig. 5 c,f). Despite some good online scores, most algorithms lack accuracy (Fig. 5 c,f). As to the long-run results, CBN-PSO proves most successful on f_{Rn} (Figs. 8-9, right, and 10 c,f). Fig. 7 exemplary shows the averaged histograms for the most successful algorithms on 30-D F_{Rn}. On the Fletcher-Powell problem as well as rotated Levy's, results look similar to f_{Rs}, meaning that the sequential methods perform best, followed by CBN-PSO. On the rotated L-function, results were similar to f_S and f_{Rn}, with superior performance of CBN-PSO.

Table II shows numerical values for the mean refined scores after the long run experiments on f_{Rs}, f_S and f_{Rn} in 10 and 30 dimensions. In addition, the method's ranks per benchmark are noted. Both ranks and scores are averaged for a condensed view. We attribute the success of CBN-PSO mainly to two facts: Firstly, PSO is known to work well even with relatively small populations, which will, on the other hand, be a major handicap for DE in the CBN-DE variant. Secondly, PSO has a relatively good exploratory behavior, which is also steered towards new areas whenever a sub-swarm gets reinitialized.

The CBN-GA algorithm is competitive in some cases, however it often shows inferior accuracy. The INMS approach, as a sequential variant of local search, does not scale well on more complex functions. NichePSO, too, is not competitive considering the scalability criteria, as expected earlier. And although ANPSO is competitive in some cases, it does not perform well in general without boosting the swarm size. Because ANPSO controls its niche radius based on swarm diversity, which is disturbed by the niche reinitialization employed, a more specialized mechanism might have to be considered. On the downside, this would add more complexity compared to relatively simple approaches such as CBN-PSO.

While CMA-ES is highly efficient on functions with large-scale correlations, such as F_{Rs}, it seems less apt on multi-funnel functions, a finding also reported in [30], for example. The clustering-based CMA-ES did not perform well, possibly due to the clustering step interfering with the self-adaptive CMA method. Still, good results on some benchmarks indicate sequential CMA-ES to be a notable candidate as a scalable MMO method, and a worthy task lies in a comparison of the behavior of the dynamic niching ES framework [14] considering the multimodal scalability criteria. The question on how to pre-select the number of expected niches could be answered similarly to the IPOP-idea, by increasing it automatically during a run. This would lead to ever-more explorative phases in the niching ES run. In analogy, any semi-

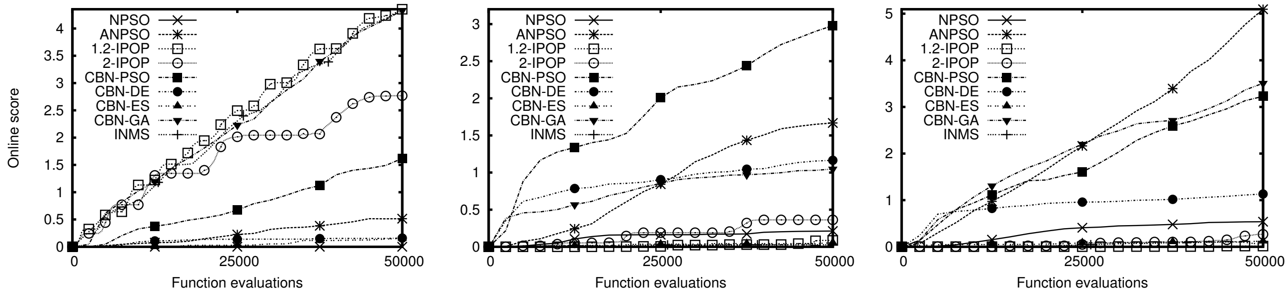


Fig. 3. Online scores on 10-D for F_{Rs}, F_S and F_{Rn}.

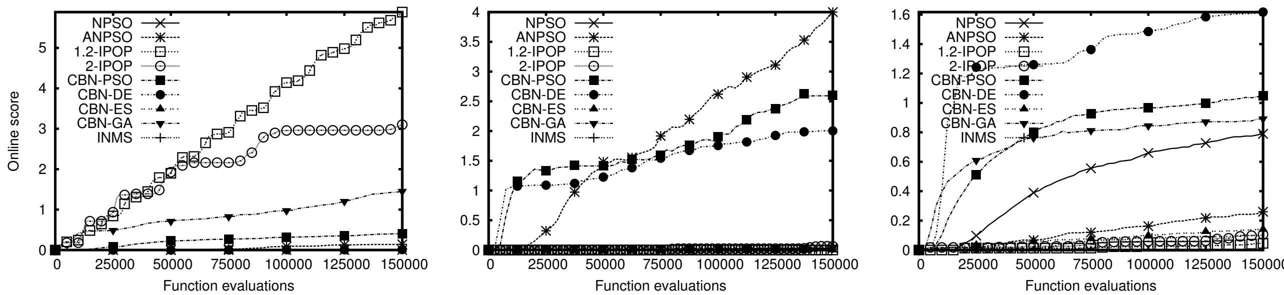
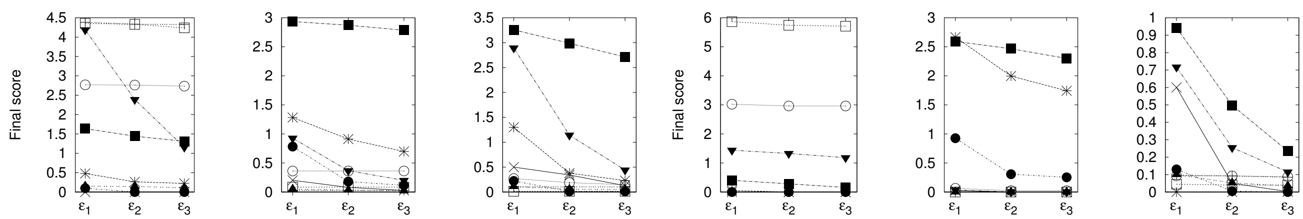


Fig. 4. Online scores on 30-D for F_{Rs}, F_S and F_{Rn}.



(a) 10-D F_{Rs} (b) 10-D F_S (c) 10-D F_{Rn} (d) 30-D F_{Rs} (e) 30-D F_S (f) 30-D F_{Rn}

Fig. 5. Final refined scores for F_{Rs}, F_S and F_{Rn} for 10-D and 30-D with increasing accuracy; $\epsilon \in \{0.01, 0.001, 0.0001\}$.

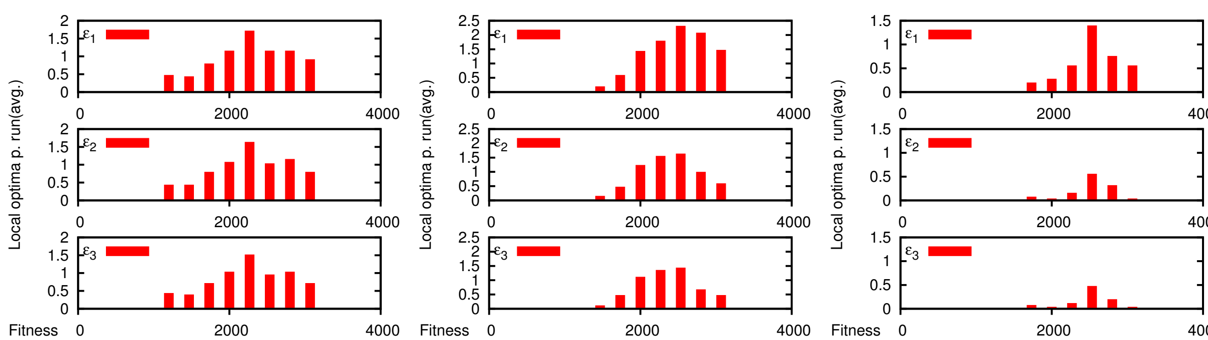


Fig. 6. Averaged final fitness histograms on F_S in 30D of CBN-PSO (left), ANPSO (middle) and CBN-DE (right).

sequential niching variant might benefit from the concept of shifting weights between exploration and exploitation during a run, so as to arrive at a diverse set of potential local optima. For example, the convergence-restart criterion in semi-sequential methods is critical. With a dynamic restart criterion depending on the quality of local optima already identified earlier during a run, a CBN-EA could invest more exploitative effort later in the run to increase the probability of finding better optima. For CBN-PSO, this can be connected to the minimal sub-population size and the clustering parameter σ. Varying σ may also serve as an approach to solve the niche-radius problem.

IV. Summary

Niching methods are dedicated to the problem of finding multiple high quality solutions of a complex objective function within a single optimization run. Many niching methods have been developed and tested on functions of very low dimensionality. The scalability to higher dimensions has seldom been

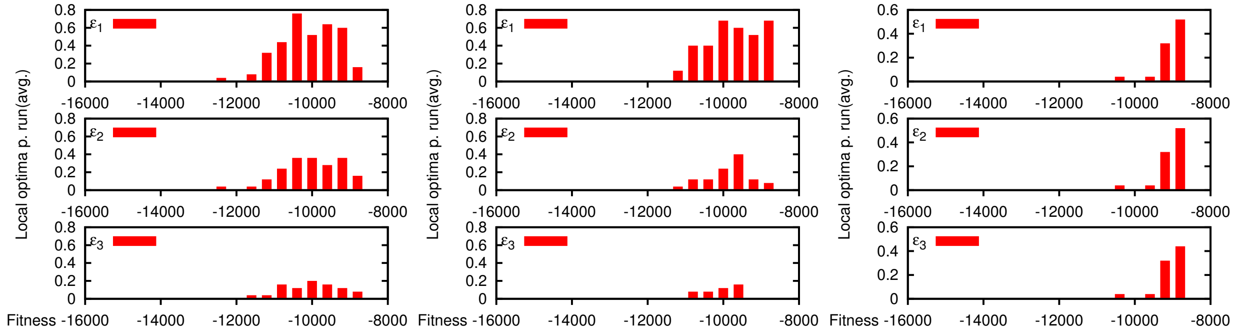


Fig. 7. Averaged final fitness histograms on F_{Rn} in 30D of CBN-PSO (left), CBN-GA (middle) and 2-IPOP-CMA-ES (right).

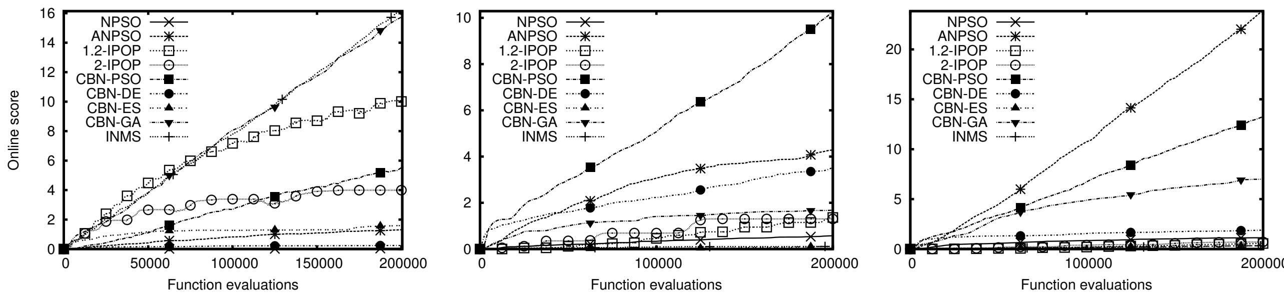


Fig. 8. Long-run online scores on 10-D for F_{Rs}, F_S and F_{Rn}.

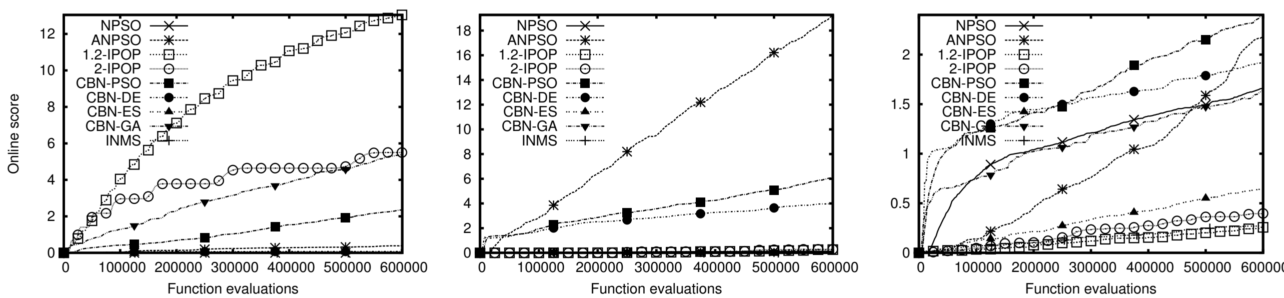
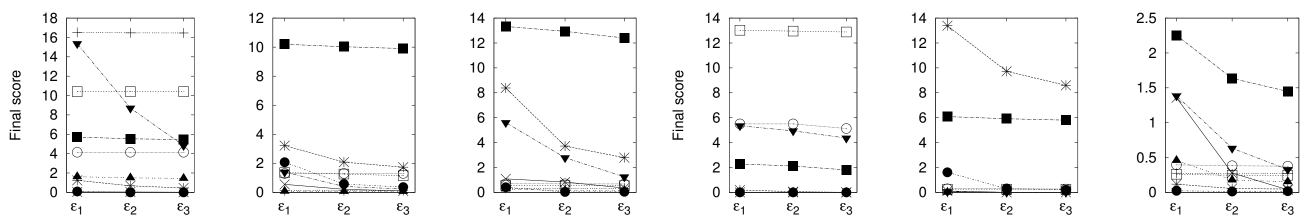


Fig. 9. Long-run online scores on 30-D for F_{Rs}, F_S and F_{Rn}.



(a) 10-D F_{Rs} (b) 10-D F_S (c) 10-D F_{Rn} (d) 30-D F_{Rs} (e) 30-D F_S (f) 30-D F_{Rn}

Fig. 10. Long-run final refined scores for F_{Rs}, F_S and F_{Rn} for 10-D and 30-D with increasing accuracy; $\epsilon \in \{0.01, 0.001, 0.0001\}$.

discussed, and if so, mostly under the assumptions that the number of local optima is known and the population size of the optimization method can be boosted accordingly.

This work introduced new criteria for the scalability of niching methods, consisting mainly of the idea that a scalable optimizer should – without external preconfiguration – deliver an increasing amount of local optima with an increasing number of target function evaluations. This may also be interpreted as an *anytime* property for metaheuristic, multimodal optimization. For a comparison of niching methods considering these properties, a scoring method based on a fitness threshold and clustered refining was employed. Using this score, several sequential and semi-sequential niching methods were analyzed with respect to the scalability criteria, including iterative Nelder-Mead-Simplex, IPOP-CMA-ES, NichePSO and its extension ANPSO, as well as four clustering-based approaches, namely CBN-ES, CBN-GA, CBN-DE, and CBN-PSO.

On the considered benchmarks, CBN-PSO showed consis-

TABLE II

OVERVIEW OF THE LONG-RUN REFINED SCORES AND ALGORITHM RANKING.

	NPSO		ANPSO		1.2-IPOP		2-IPOP		CBN-PSO		CBN-DE		CBN-ES		CBN-GA		INMS	
	Sc	R	Sc	R	Sc	R	Sc	R	Sc	R	Sc	R	Sc	R	Sc	R	Sc	R
F_{Rs}, 10-D	.01	8	.45	7	10.40	2	4.14	5	5.45	3	0	9	1.44	6	4.83	4	**16.46**	**1**
F_{Rs}, 30-D	0	6	0	6	**12.88**	**1**	5.13	2	1.80	4	0	6	0	6	4.36	3	.01	5
F_S, 10-D	.09	8	1.73	2	1.15	4	1.30	3	**9.90**	**1**	.36	5	.06	9	.23	6	.11	7
F_S, 30-D	0	6	**8.60**	**1**	.25	4	.26	3	5.80	2	.20	5	0	6	0	6	0	6
F_{Rn}, 10-D	.35	6	2.79	2	.54	5	.67	4	**12.40**	**1**	.04	9	.12	8	1.24	3	.32	7
F_{Rn}, 30-D	.03	8	.04	7	.24	5	.37	2	**1.44**	**1**	.01	9	.15	6	.32	3	.27	4
Mean rank	7.00		4.17		3.50		3.17		**2.00**		7.16		6.83		4.17		5.00	
Mean score	0.08		2.27		4.24		1.98		**6.13**		0.10		0.30		1.83		2.86	

tent scalability with good accuracy. CBN-GA, ANPSO and IPOP-CMA-ES showed scalable behavior in some cases, with IPOP-CMA-ES being generally the most accurate method. We assume that the success of CBN-PSO mainly comes from PSO's potential to work well even with rather small population sizes, which are prone to come up in parallel niching due to the distribution of search capacity among several niches.

For future work, the evaluation of other current niching methods is considered. Moreover, an extension of CBN-PSO to dynamically change its exploration-exploitation behavior would aim at increasing the probability of finding not just more, but also better local optima with increasing runtime.

REFERENCES

[1] O. Shir, C. Siedschlag, T. Bäck, and M. Vrakking, "Niching in evolution strategies and its application to laser pulse shaping," *Lecture Notes in Computer Science*, vol. 3871, pp. 85–96, 2006.

[2] M. Kronfeld, A. Dräger, M. Aschoff, and A. Zell, "On the benefits of multimodal optimization for metabolic network modeling," in *Proc. of the German Conf. on Bioinformatics 2009* (I. Grosse *et al.*, eds.), vol. P-157 of *Lecture Notes in Informatics*, pp. 191–200, GI Society, 2009.

[3] A. Engelbrecht, *Computational Intelligence: An Introduction*. New York, NY, USA: Halsted Press, 2002.

[4] F. Streichert, G. Stein, H. Ulmer, and A. Zell, "A clustering based niching EA for multimodal search spaces," in *In Proceedings of Evolution Artificielle (LNCS 2935)*, pp. 293–304, Springer-Verlag, 2003.

[5] K. A. de Jong, *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, Uni. of Michigan, Ann Arbor, MI, USA, 1975.

[6] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Proc. of the Second International Conference on Genetic Algorithms and their Applications*, pp. 41–49, Lawrence Erlbaum Associates, Inc., 1987.

[7] A. Pétrowski, "A clearing procedure as a niching method for genetic algorithms," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pp. 798–803, 1996.

[8] S. W. Mahfoud, "Crowding and preselection revisited," in *Parallel Problem Solving from Nature 2* (R. Männer and B. Manderick, eds.), pp. 27–36, Elsevier, 1992.

[9] G. Harik, "Finding multimodal solutions using restricted tournament selection," in *Proceedings of the Sixth International Conference on Genetic Algorithms*, pp. 24–31, Morgan Kaufmann, 1995.

[10] D. Beasley, D. Bull, and R. Martin, "A sequential niche technique for multimodal function optimization," *Evolutionary computation*, vol. 1, no. 2, pp. 101–125, 1993.

[11] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. on Evolutionary Computation*, vol. 8 (3), pp. 211–224, 2004.

[12] S. Tsutsui, Y. Fujimoto, and A. Ghosh, "Forking genetic algorithms: GAs with search space division schemes," *Evolutionary computation*, vol. 5, no. 1, pp. 61–80, 1997.

[13] R. Ursem, "Multinational evolutionary algorithms," in *Congress on Evolutionary Computation (CEC)*, vol. 3, pp. 1633–1640, 1999.

[14] O. Shir and T. Bäck, "Niching in evolution strategies," in *Proc. of the 2005 Conference on Genetic and Evolutionary Computation (GECCO)*, p. 916, ACM, 2005.

[15] R. Thomsen, "Multimodal optimization using crowding-based differential evolution," in *Congress on Evolutionary Computation (CEC)*, vol. 2, pp. 1382–1389, 2004.

[16] R. Brits, A. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimizer," in *Proceedings of the Fourth Asia-Pacific Conference on Simulated Evolution and Learning*, pp. 692–696, 2002.

[17] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," in *Proc. of the 8th annual conference on Genetic and evolutionary computation (GECCO)*, (New York, NY, USA), pp. 3–10, ACM, 2006.

[18] R. Brits, A. Engelbrecht, and F. van den Bergh, "Scalability of Niche PSO," in *The 2003 IEEE Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of*, pp. 228 – 234, 2003.

[19] R. Brits, A. Engelbrecht, and F. Van den Bergh, "Locating multiple optima using particle swarm optimization," *Applied Mathematics and Computation*, vol. 189(2), pp. 1859–1883, 2007.

[20] G. Singh and D. Kalyanmoy Deb, "Comparison of multi-modal optimization algorithms based on evolutionary algorithms," in Proc. of the 8th annual conference on Genetic and evolutionary computation (GECCO), (New York, NY, USA), pp. 1305–1312, ACM, 2006.

[21] J. A. Nelder and R. Mead, "A simplex method for function minimization," *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[22] A. Auger and N. Hansen, "A restart CMA evolution strategy with increasing population size," in *Congress on Evolutionary Computation (CEC* (B. McKay *et al.*, eds.), vol. 2, pp. 1769–1776, 2005.

[23] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE Int. Conf. on Neural Networks*, (Perth, Australia), 1995.

[25] J. H. Holland, *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[26] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[27] O. M. Shir and T. Bäck, "Niche radius adaptation in the CMA-ES niching algorithm," in *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, vol. 4193, pp. 142–151, Springer, 2006.

[28] A. Levy and A. Montalvo, "The tunneling algorithm for the global minimization of functions," *SIAM Journal on Scientific and Statistical Computing*, vol. 6, p. 15, 1985.

[29] P. N. Suganthan, N. Hansen, J. Liang, K. Deb, Y. Chen, A. Auger, and A. Tiwari, "Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization," Tech. Rep. 2005005, Nanyang Technological University, Singapore, 2005.

[30] M. Lunacek, D. Whitley, and A. Sutton, "The impact of global structure on search," *Parallel Problem Solvng from Nature – PPSN X*, vol. 5199, pp. 498–507, 2008.