

NICHING PROSPECTS

Mike Preuss

Algorithm Engineering,

Universität Dortmund

mike.preuss@uni-dortmund.de

Abstract Although a large number of evolutionary algorithms have been proposed to efficiently treat multimodal problems, it is currently unclear under what conditions they can be faster than iterated local search algorithms. We tackle this question, assuming we had means to efficiently and errorlessly determine the corresponding basin of attraction for each individual (basin identification) by employing a simplified niching model EA that avoids superfluous local searches. Monte Carlo simulations show that outperforming the iterated local search is possible but difficult; the expected speedup is rather low if basins are approximately equally sized.

Keywords: Niching evolutionary algorithms, Basin model, Monte Carlo simulations

1. Introduction

Niching in *evolutionary algorithms* (EA) appears to be a heterogeneous collection of techniques applied to enhance their ability to cope with multimodal objective functions by implementing some form of parallelization, either in terms of search space or time, or both. Does it contain all EA variants suggested for multimodal optimization? Surely not. But to state what exactly distinguishes niching approaches from other ones seems non-trivial, as—despite existing, partly contradictory definitions—the *evolutionary computation* (EC) community apparently does not yet possess a unified taxonomic view on multimodal evolutionary optimization. It is our aim to contribute to a movement into this direction by investigating what niching actually is and what it can do to improve evolutionary algorithms.

1.1 Niching Definitions

Out of the large set of publications dealing with niching or similar techniques in EC (e.g. deJong [4] and Goldberg [7] as some of the earliest) we select only two opinions to show where to locate possible disagree-

ments in defining niching. Mahfoud [12] gives the following functional specification of niching methods in an optimization context (p. 61):

The litmus test for a niching method, therefore, will be whether it possesses the capability to find multiple, final solutions within a reasonable amount of time, and to maintain them for an extended period of time.

He additionally states that the multiple solutions correspond to multiple local optimizers. Beyer et al. [6] include the process of separation, too. However, they also add diversity maintenance in their definition:

Niching—process of separation of individuals according to their states in the search space or maintenance of diversity by appropriate techniques, e.g. local population models, fitness sharing, or distributed EA

Whenever speaking of niches in EAs for static black box optimization, authors usually identify them with basins of attraction, at least for real-valued optimization. As Mahfoud points out, diversity maintenance is related to niching but must not be pursued too rigorously because it enables, but does not guarantee finding many basins, depending on the basin distribution within search space. In this sense, combining parts of both specifications, referring to basins of attraction, and leaving out diversity maintenance leads us to the following new definition:

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.

Undoubtedly, all EAs have local search capabilities. Therefore, it must be the process of locating basins that induces difficulties and requires experimentation with many EA variants to establish niching. In accordance to the explicit/implicit diversity maintenance scheme suggested by Eiben and Smith [5], we further partition niching EAs into two groups, performing explicit or implicit *basin identification*. Explicit basin identification methods—detecting the basin of each individual—divide the individuals into subpopulations, according to their basins.

1.2 Existing Approaches

Over the last 30 years, a large variety of niching techniques has been suggested. Comprehensive comparative studies are rare, but the existing (e.g. Mahfoud [11], Watson [23]) give hints on the relation between fitness landscape properties and performance of different niching methods. However, despite several recent approaches (Beasley et al. [2], Pétrowski [16, 17], Jelasity [8], Ursem [21], Wineberg [24], Li et al. [9], Streichert et al. [20], Shir [19], Ando et al. [1]), in the face of a multi-

tude of possibilities one is tempted to resort to the 'traditional' methods *crowding* [4] and *sharing* [7], or variations thereof. What is the reason for this dissatisfactory tentativeness? There may be several, e.g.:

- The diverse character of the proposed methods, algorithmically as well as in descent from different origins, complicates gathering a viable overview. Available results are not directly comparable.
- Only few taxonomic attempts exist for multimodal EAs, and the existing ones by Ursem [22], and Eiben and Smith [5] utilize different, mutually incomparable criteria, as avoid/repair strategy in the former and diversity maintenance in the latter case.
- The aspired task is not concrete enough or unreachable.

As the stream of new methods does not cease, one may ask what the motivation behind designing new niching EAs is. The seemingly underlying, yet unreached aim is to convincingly beat one of the simplest algorithms for multimodal objective functions, the iterated/parallelized hillclimber/local search. According to the NFL, this task is venturous when optimizing general multimodal problems, but it may be possible for problem classes exhibiting certain exploitable properties.

1.3 Biological Background

Importing concepts from biology (ecology), which undoubtedly is the origin of the general idea of niching for EAs, appears problematic. Biologists now tend to view separation into niches as a process the affected living beings actively take part in, also treated as *niche construction*, Odling-Smee et al. [15]. Whereas individuals in canonical EAs are merely collections of values without a 'life of their own', living beings act on highly dynamic fitness landscapes and must pursue several objectives (e.g. food and reproduction).

The related problem of speciation—the term species often denotes separate subpopulations in niching EAs—currently is one of the most progressive research topics in evolutionary biology, with Mayr's reproductively isolated populations [13, 14] and the allopatric (geographic) speciation mechanism as predominant concepts. Although these two can be (and are) adapted for use in EAs, biologists are still far from having reached consensus concerning all issues raised with the problem of speciation, and thus not able to provide a proper foundation to argue on in the EC field. The current state of the speciation debate is summarized in Coyne and Orr [3]. As an example for a controversially discussed yet unsolved problem, we name the formation and maintenance of sexual reproduction. This issue is dismissed in EA research, in favor of asexual

populations, for which in turn no widely accepted speciation concept exists in biology. In consequence, biological terms shall be used with extreme care when applied to niching EAs to prevent conceiving meanings where there are only metaphors.

2. Aims and Methods

In the following, our main task is to gather evidence in favor of or against the (in EC) prevalent belief that niching EAs can outperform iterated local search (ILS, see Lourenco et al. [10]) algorithms. Note that this is an existential precondition for designing further niching EAs as these are usually algorithmically much more complex. We thus do the second step prior to the first and simply assume the existence of an efficient basin identification method for population based EAs. This would enable deciding if any two individuals are located in the same basin or not. The first question to investigate thus is:

- Given that basin identification works, how much faster can a niching EA be in terms of a *redundancy factor* (measuring superfluous local searches, see Beasley et al. [2]), compared to ILS algorithms?

We employ a very simple niching model EA and estimate the amount of local searches needed for reasonable basin numbers and population sizes by means of Monte Carlo simulations.

3. Simplistic Niching Model EA

Modeling the behavior of a real niching EA on an idealized multimodal objective function still bears enormous complexity. The whole local search process in the detected basins must be considered, and is likely to heavily depend on algorithm and problem parameters.

Hence, for our niching model EA, we choose the single local search as unit of measurement. We further assume that for any (start) population of search points, a basin identification method exists that returns an errorless search point to basin mapping in negligible time. This condition describes an optimal situation—for any real niching EA, basin identification will require computational effort. Additionally, it may not be possible to detect the basin of an individual as soon as it enters it. Thus, the implied advantage of our ideal niching EA which consists of breaking unnecessary local searches at the start may not be realizable in full. But, unless other techniques are applied to reduce the optimization effort (e.g. utilization of attained information to speed up subsequent local searches), *any* niching EA can not be faster in terms of local searches than the niching model EA—we obtain an estimation for a lower bound.

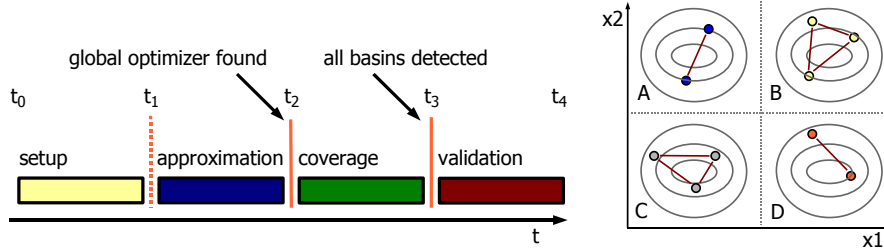


Figure 1. Left: Four phases of a heuristic optimization process. We are interested in detecting t_2 and t_3 . Right: Niching model EA population after initialization and basin identification. Individuals residing in one basin are connected by lines.

In a real niching EA, the number of covered c of a total of b basins for a randomly initialized start population would fluctuate according to population size and basin distribution. However, we set it constant to simplify studying the effect of saving local searches. Summarizing, the model is based on the following assumptions:

- Basin identification is perfect and has zero cost.
- Local searches always succeed and have equal cost of 1.
- Any start population covers exactly c of b existing basins.

Without basin identification, one is thrown back to iterated/parallelized local searches for which the required effort is known [2]. Covering the whole basin set with randomly initialized local searches results in a relative local search overhead, measured by the redundancy factor R :

$$R = \sum_{i=1}^b \frac{1}{i} \stackrel{b>3}{\approx} \gamma + \ln b \quad (1)$$

For entering each of the b basins at least once, $R \times b$ local searches are necessary on average. Here, $\gamma \approx 0.577$ is the Euler-Mascheroni constant.

Instead of conducting single local searches, the niching model EA repeatedly starts with a randomly initialized set of individuals and performs only necessary local searches until all basins have been visited (Fig. 2). We do not specify how the local searches are implemented; they may be realized e.g. by mating restrictions, or separate populations, or embedded helper methods. Note that basin identification only needs to detect if individuals are located in the same basin; it is not required to properly recognize each basin as such (Fig. 1, right).

What kind of performance data do niching model EA simulation runs deliver? Figure 1 (left) displays the phases of any heuristic optimization

```

niching model EA {
  repeat {
    randomly initialize individuals on  $c$  of  $b$  basins;
    basin identification: match individuals to basins;
    select one individual per basin =  $c$  individuals;
    perform  $c$  local searches on selected individuals;
  } until stopped externally (all basins visited);
}

```

Figure 2. Niching model EA in pseudo-code.

algorithm in terms of basin detection. During setup, the algorithm is prepared and started and yields the first result in t_1 . This approximation phase lasts until the global optimizer is hit the first time at t_2 . It shall be noted that especially in real-world applications, this point is often never reached because evaluations may be too costly. The coverage phase is needed to visit each basin at least once and ends with t_3 . Unless the number of basins is known in advance, it seems impossible to determine t_3 from inside an optimization algorithm. It is up to the user to stop it when no new information can be obtained from running further (t_4). In case of the niching model EA, t_2 and t_3 are measurable because the basin set is known. Note that the redundancy factor stated in Eqn. 1 is equivalent to t_3 which thus not refers to the expected first hitting time but to the end of the coverage phase.

In the following, we present two experiments in order to investigate the influence of basin number b and coverage/parallelized searches c on t_2 and t_3 for the niching model EA. Firstly, equally sized basins are studied. Secondly, we review occurring changes for unequally sized basins.

Experiment 1: Global optimizer/coverage detection times, equal basins.

Pre-experimental planning: The appropriate number of repeats is determined to 10000 during first tests; relative standard deviations are thus decreased well below 1%.

Task: Measure t_2 and t_3 and detect how they relate to the number of basins b and parallel searches (covered basins) c .

Setup: We simulate all $b, c \in \{1, 2, \dots, 50\} : b \geq c$ with 10000 repeats each. Probabilities for encountering any basin during random initialization are equal and set to $\frac{1}{b}$.

Experimentation/Visualization: Figure 3 depicts averaged measures for t_3 (left) and t_2 (right).

Observations: Whereas the number of parallel searches c clearly affects t_3 , it seems to lack any influence on t_2 which only depends on the number

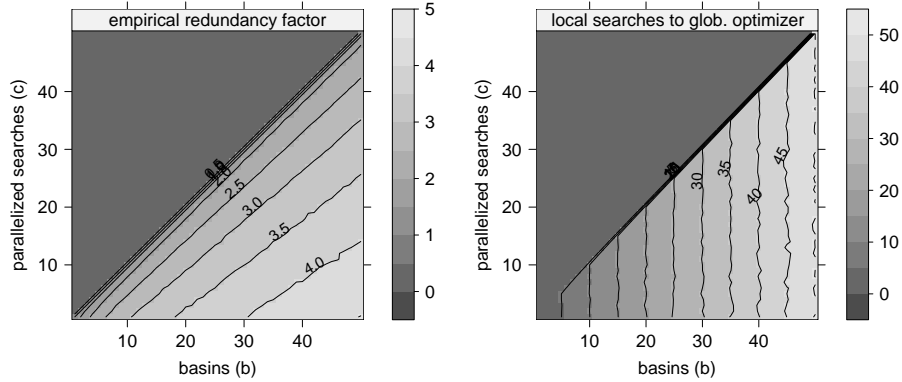
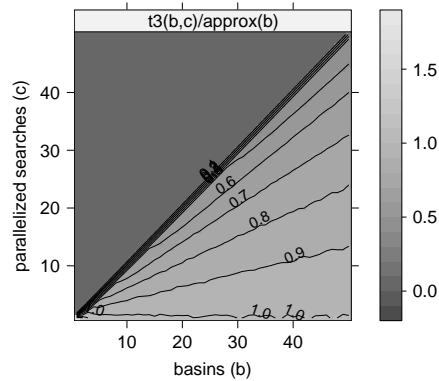


Figure 3. Left: Measured redundancy factors (t_3), right: local searches needed to locate the global optimum (t_2). Both are averaged from 10000 simulations per point.

of basins b ($\mathbb{E}(t_2) = b$). To clarify the influence of c on t_3 , we picture measured t_3 , divided by the approximation given by Eqn. 1 (Fig. 4).

Fig. 4. Measured redundancy factor (t_3) as fraction of the approximation for repeated single local searches (Eqn. 1) for the same number of basins b . If $\frac{c}{b} \leq 0.9$, the coefficient of the observed linear relation is similar to $\sqrt[3]{1 - \frac{c}{b}}$ (by visual comparison), resulting in the approximation $t_3(b, c) \approx \sqrt[3]{1 - \frac{c}{b}} \cdot (\gamma + \ln b)$.



Discussion: Different values for c do not change t_2 at all, meaning that parallel searches do not increase or decrease the expected time needed to arrive at the global optimum. Approaches targeting at this effect for approximately equally sized basins are thus doomed to fail. Nevertheless, the amount of local searches needed for complete coverage (t_3) is reduced for $c > 1$. However, the save is small and the utilized basin identification technique must be very efficient not to loose it again.

Experiment 2: Detect t_2 and t_3 for unequally sized basins.

Pre-experimental planning: The maximum size difference was fixed to 10 as first experiments indicate a sufficient change in obtained results.

Task: Similar to Exp. 1.

Setup: Similar to Exp. 1, but with uniform randomly distributed relative basin sizes between 1.0 and 10.0.

Experimentation/Visualization: Averaged t_2 and t_3 measures are depicted in Fig. 5.

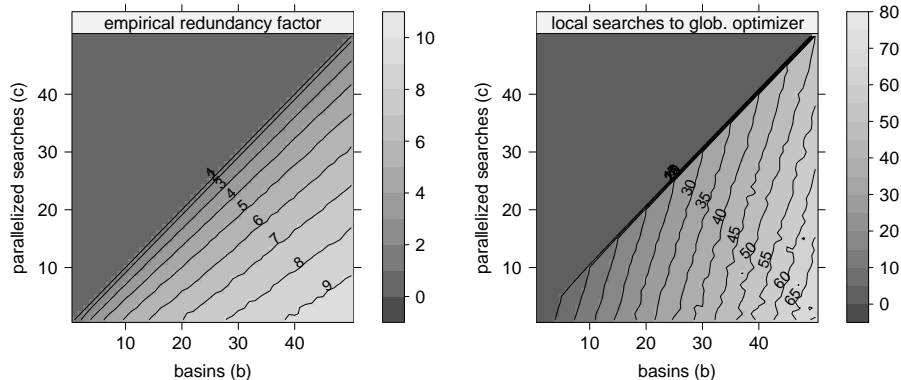


Figure 5. Redundancy factors (t_3 , left) and local searches to hit the global optimizer (t_2 , right), averages of 10000 simulations. Relative basin size sizes are 1 to 10.

Observations: Firstly, measured values for t_3 arrive at much higher values than for the case of equal basins. Secondly, the growth rate on the basin axis ($c = 1$) appears to be between logarithmic and linear—compared to logarithmic in Exp. 1. In contrast to the findings of Exp. 1, t_2 now is affected by changing values of c . For $b = c$, that is, all basins are covered by the parallel search, $t_2 = b$ still holds. But the lower $\frac{c}{b}$ is, the larger t_2 gets.

Discussion: Obviously, optimization gets harder if basins are unequally sized. This is well in accordance with expectation. Now, t_2 and t_3 both depend on $\frac{c}{b}$. We may conclude that larger relative basin size differences lead to larger potential performance advantages of niching EAs. On the other hand, basin identification probably gets harder, too.

4. Conclusions

Previous studies (e.g. Preuss et al. [18]) have shown that canonical EAs are not well suited for multimodal optimization. Are niching EAs? According to our simulations, there is some exploitable potential, but it is small for equally sized basins. It appears that chances are getting better the larger basin size differences are. However, we assumed existence of an efficient basin identification method, which utilizes population topologies in search space and thus depends on the number of dimensions of a problem. Whether and for what problems such technique can be fast enough to enable outperforming an ILS still remains to be seen.

References

- [1] S. Ando, E. Suzuki, and S. Kobayashi. Sample-based Crowding Method for Multimodal Optimization in Continuous Domain. In G. W. Greenwood, editor, *Proc. 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, UK*, Piscataway NJ, 2005. IEEE Press.
- [2] D. Beasley, D. R. Bull, and R. R. Martin. A sequential niche technique for multimodal function optimization. *Evolutionary Computation*, 1(2):101–125, 1993.
- [3] J. A. Coyne and H. A. Orr. *Speciation*. Sinauer Associates, Inc., Sunderland, MA, 2004.
- [4] K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.
- [5] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, Heidelberg, New York, 2003.
- [6] H.-G. B. et al. Evolutionary algorithms - terms and definitions. VDI/VDE guideline 3550, leaf 3, 2003. online version at <http://ls11-www.cs.uni-dortmund.de/people/beyer/EA-glossary/def-engl-html.html>.
- [7] D. E. Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In *Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application*, pages 41–49, Mahwah, NJ, USA, 1987. Lawrence Erlbaum Associates, Inc.
- [8] M. Jelasity. Uego, an abstract niching technique for global optimization. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *PPSN*, volume 1498 of *Lecture Notes in Computer Science*, pages 378–387. Springer, 1998.
- [9] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, 10(3):207–234, 2002.
- [10] H. R. Lourenco, O. Martin, and T. Stützle. Iterated local search. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*. Kluwer, 2002.
- [11] S. W. Mahfoud. A comparison of parallel and sequential niching methods. In L. Eshelman, editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 136–143, San Francisco, CA, 1995. Morgan Kaufmann.

- [12] S. W. Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, IL, 1995.
- [13] E. Mayr. *Systematics and the Origin of Species*. Columbia University Press, New York, 1942.
- [14] E. Mayr. Species, classification, and evolution. In R. Arai, M. Kato, and Y. Doi, editors, *Biodiversity and Evolution*. National Science Museum Foundation, Tokyo, 1995.
- [15] F. J. Odling-Smee, K. N. Laland, and M. W. Feldman. *Niche Construction—The neglected process in evolution*. Princeton University Press, Princeton and Oxford, 2003.
- [16] A. Pérowski. A clearing procedure as a niching method for genetic algorithms. In T. Fukuda, T. Furuhashi, and D. B. Fogel, editors, *Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC '96), Nagoya*, pages 798–803, Piscataway NJ, 1996. IEEE Press.
- [17] A. Pérowski. A Classification Tree for Speciation. In P. J. Angeline and V. W. Porto, editors, *Proc. 1999 Congress on Evolutionary Computation (CEC'99), Washington D.C.*, pages 204–211, Piscataway NJ, 1999. IEEE Press.
- [18] M. Preuss, L. Schönemann, and M. Emmerich. Counteracting genetic drift and disruptive recombination in $(\mu^+ \lambda)$ -ea on multimodal fitness landscapes. In H.-G. Beyer, editor, *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 865–872, New York, NY, USA, 2005. ACM Press.
- [19] O. M. Shir. Niching in evolution strategies. In H.-G. Beyer, editor, *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 865–872, New York, NY, USA, 2005. ACM Press.
- [20] F. Streichert, G. Stein, H. Ulmer, and A. Zell. A clustering based niching method for evolutionary algorithms. In E. Cantú-Paz, editor, *Genetic and Evolutionary Computation – GECCO-2003*, pages 644–645, Berlin, 2003. Springer-Verlag.
- [21] R. K. Ursem. Multinational evolutionary algorithms. In P. J. Angeline, editor, *Proceedings of the Congress of Evolutionary Computation (CEC-99)*, volume 3, pages 1633–1640, Piscataway, NJ, 1999. IEEE Press.
- [22] R. K. Ursem. *Models for Evolutionary Algorithms and Their Applications in System Identification and Control Optimization*. PhD thesis, EVALife, Department of Computer Science, University of Aarhus, 2003.
- [23] J.-P. Watson. A performance assessment of modern niching methods for parameter optimization problems. In W. Banzhaf et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference*, volume 1, pages 702–709, Orlando, Florida, USA, 13–17 1999. Morgan Kaufmann.
- [24] M. Wineberg. *Improving the Behavior of the Genetic Algorithm in a Dynamic Environment*. PhD thesis, Carleton University, Ottawa, Canada, 2000.