Chapter 10
Clustering-Based Niching

10.1 Introduction

Some optimization problems posses many potential locations of local and global optima. The potential locations are often denoted as basins in solution space. In many optimization scenarios, it is reasonable to evolve multiple equivalent solutions, as one solution may not be realizable in practice. Alternative optima allow the practitioner the fast switching between solutions. Various techniques allow the maintenance of diversity that is necessary to approximate optima in various basins of solution spaces. Such techniques are, e.g., large populations, restart strategies, and niching. The latter is based on the detection of basins and simultaneous optimization within each basin. Hence, niching approaches implement two important steps: (1) the detection of potential niches, i.e., parts of solution space that may accommodate local optima and (2) the maintenance of potentially promising niches to allow convergence of the optimization processes within each niche.

In this chapter, we propose a method to detect multiple locations in solution space that potentially accommodate good local or even global optima for ES [1]. This detection of basins is achieved by sampling in solution space, selecting the best solutions w.r.t. their fitness, and then detecting potential niching locations with clustering. For clustering, we apply DBSCAN [2], which does not require the initial specification of the number of clusters, and k-means that successively repeats cluster assignments and cluster mean computations.

This chapter is structured as follows. Section 10.2 gives a short introduction to clustering concentrating on DBSCAN, k-means, and the Dunn index to evaluate clustering results. Section 10.3 introduces the clustering-based niching concept. Related work is introduced in Sect. 10.4. Experimental results are presented in Sect. 10.5. Last, Sect. 10.6 summarizes the most important findings.

O. Kramer, *Machine Learning for Evolution Strategies*,
Studies in Big Data 20, DOI 10.1007/978-3-319-33383-0_10

10.2 Clustering

Clustering is the task of grouping patterns without label information. Given patterns $\mathbf{x}_i$ with $i = 1, \ldots, N$, the task is to group them into clusters. Clustering aims at maximizing the homogeneity among patterns in the same cluster and the heterogeneity of patterns in different clusters. Various evaluation criteria for clustering have been presented in the past, e.g., the DUNN index [3] that we use in the experimental part. Some techniques require the specification of the number of clusters at the beginning, e.g., k-means, which is a prominent clustering method. DBSCAN and k-means will shortly be sketched in the following.

DBSCAN [2] is a density-based clustering method. With a user-defined radius `eps` and number `min_samples` of patterns within this radius, DBSCAN determines core points of clusters, see Fig. 10.1. Core points lie within regions of high pattern density. DBSCAN assumes that neighboring core points belong to one cluster. Using the core points, the cluster is expanded and further points within radius `eps` are analyzed. All core points that are reachable from a core point belong to the same cluster. Corner points are points that are reachable from a core point, but that are not core points themselves. Patterns that are neither core points nor corner points are classified as noise.

For comparison, we experiment with the famous clustering method k-means. In k-means, a number k of potential clusters has to be detected before the clustering process begins. First, k cluster centers are randomly initialized. Then, the two steps of assigning patterns to the nearest cluster centers and computing the new cluster centers with the assigned patterns, are iteratively repeated until the movements of the cluster centers fall below a threshold value.

Cluster evaluation measures are often based on inter- and intra-cluster variance. A famous clustering measure is the DUNN index. It computes the ratio between the distance of the two closest clusters and the maximum diameter of all clusters, for an illustration see Fig. 10.2. Let $c(\mathbf{x}_i)$ be a function that delivers the cluster pattern $\mathbf{x}_i$ is assigned to. The minimum distance between all clusters is defined as

$$\delta = \min_{i,j,i \neq j, c(\mathbf{x}_i) \neq c(\mathbf{x}_j)} \|\mathbf{x}_i - \mathbf{x}_j\|_2. \tag{10.1}$$

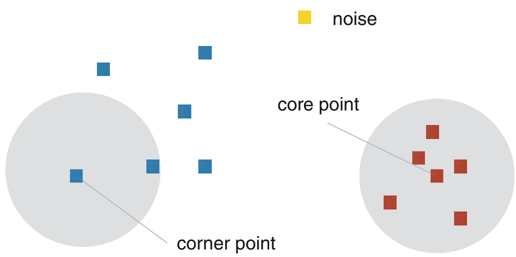

Fig. 10.1 Illustration of core and corner points of DBSCAN

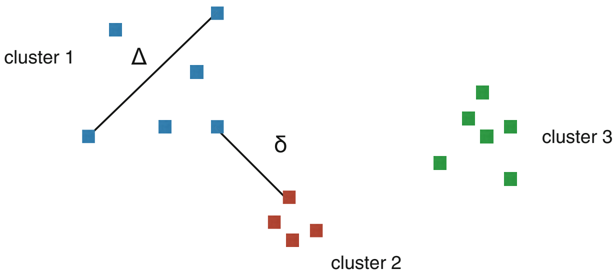

Fig. 10.2 Illustration of DUNN index

The maximal cluster diameter is defined as

$$\Delta = \max_{i,j,i \neq j, c(\mathbf{x}_i)=c(\mathbf{x}_j)} \|\mathbf{x}_i - \mathbf{x}_j\|_2. \tag{10.2}$$

The DUNN index is defined as δ/Δ and has to be maximized, i.e., small maximal cluster diameters and large minimal cluster distances are preferred. The DUNN index is useful for our purpose, as small niches and large distances between niches are advantageous for the maintenance of niches during the optimization process.

The application of DBSCAN and k-means in SCIKIT- LEARN has already been introduced in Chap. 5 and is only shortly revisited here.

- `DBSCAN(eps=0.3, min_samples=10).fit(X)` is an example, how DBSCAN is accessed for a pattern set `X`, also illustrating the use of both density parameters.
- `KMeans(n_clusters=5).fit(X)` is the corresponding example for k-means assuming 5 clusters.

10.3 Algorithm

In this section, we introduce the clustering-based niching approach. Algorithm 8 shows the pseudocode, which is denoted as NI-ES in the following. In the initialization step, the objective is to detect all niches. For this sake, the approach first samples λ' candidate solutions randomly with uniform distribution in the whole feasible solution space. This initial phase targets at exploring the solution space for the detection of basins. For this sake, each solution $\mathbf{x}_1, \ldots, \mathbf{x}_{\lambda'}$ has to be evaluated w.r.t. fitness function f. Among the λ' candidates the $N = \varphi \cdot \lambda'$ best solutions are selected. The proper choice of rate $\varphi \in (0, 1)$ depends on the structure of the optimization problem, e.g., on the number and size of basins as well as on the dimensionality d of the problem.

Algorithm 8 NI-ES

initialize $\mathbf{x}_1, \ldots, \mathbf{x}_{\lambda'}$ (random, uniform dist.)
evaluate $\{\mathbf{x}_i\}_{i=1}^{\lambda'} \rightarrow \{f(\mathbf{x}_i)\}_{i=1}^{\lambda'}$
select N best solutions
cluster $\mathbf{x}_1, \ldots, \mathbf{x}_N \rightarrow C$
for cluster in C **do**
  cluster center $\rightarrow$ initial solution $\mathbf{x}$
  intra-cluster variance $\rightarrow$ initial step size σ
  (1+1)-ES until termination condition
end for

In the next step, the remaining candidate solutions are clustered. From the selection process, basins turn out to be agglomerations of patterns in solution space that can be detected with clustering approaches. The result of the clustering process is an assignment of the N solutions to k clusters C.

For the optimization within each niche, i.e., in each cluster of C, an initial step size for the Gaussian mutation has to be determined from the size of the basins. The step size should be large enough to allow fast search within a niche, but small enough to prevent their unintentional leaving. We propose to employ the intra-cluster variance as initial step size σ. With the center of each niche as starting solution $\mathbf{x}$, k (1+1)-ES begin their search in each niche until their termination conditions are reached. This step can naturally be parallelized.

First, the process concentrates on the detection of potential niches. For this sake, a random initialization with uniform distribution is performed in solution space. The number of candidate solutions during this phase must be adapted to the dimension d of the problem. In the experimental part, we will focus on the curse of dimensionality problem. The trade-off in this step concerns the number of patterns. A large number improves the clustering result, i.e., the detection of niches, but costs numerous potentially expensive fitness function evaluations.

10.4 Related Work

Niching is a method for multimodal optimization that has a long tradition [4]. Shir and Bäck [5] propose an adaptive individual niche radius for the CMA-ES [6]. Pereira et al. [7] integrate nearest-better clustering and other heuristic extensions into the CMA-ES. Similar to our approach, their algorithm applies an exploratory initialization phase to detect niches.

Sadowski et al. [8] propose a clustering-based niching approach that takes into account linkage-learning and that is able to handle binary and real-valued objective variables including constraint handling. Preuss et al. [9] take into account properties like size relations, basins sizes, and other indicators for the identification of niches. For clustering, nearest-better clustering and Jarvis-Patrick clustering are used.

For multi-objective optimization, we employ DBSCAN to detect and approximate equivalent Pareto subsets in multi-objective optimization [10]. This approach uses explorative cluster detection at the beginning, but tries to explore niches during the usual optimization runs. Also Bandaru and Deb [11] concentrate on niching for multi-objective optimization arguing that dominance and diversity preservation inherently cause niching.

Niching is a technique that is also applied in other heuristics. Biswas et al. [12] focus on the maintenance of solutions in each niche with an information-sharing approach that allows parallel convergence in all niches. Their experimental analysis concentrates on differential evolution, but the approach can be applied to PSO and other heuristics as well.

In EDAs, restricted tournament replacement is a famous niching concept, and is applied, e.g. in [13]. It randomly samples a set of solutions from the solution space, searches the closest in the population w.r.t. a genotypic distance measure like the Euclidean distance in $\mathbb{R}^d$, and replaces the solution in the tournament set, if its fitness is worse than the fitness of the close solution. Cheng et al. [14] compare PSO algorithms with different neighborhood structures defining the particle groups for the PSO update step on the CEC 2015 single objective multi-niche optimization benchmark set. They report that the ring neighborhood structure performs best on the test set. For variable mesh optimization, Navarro et al. [15] propose a general niching framework. Niching concepts find various applications, e.g., in learning classifier systems [16] or interestingly for clustering data spaces [17].

10.5 Experimental Analysis

In the following, we experimentally analyze the NI-ES by visualizing the clustering results. We analyze the behavior of DBSCAN and k-means for clustering-based niching under different initial sampling conditions. Further, we analyze the maintenance of niches when optimizing with (1+1)-ES in each niche.

Initially, $\lambda' = 1000$ solutions are randomly sampled with uniform probability in the interval $[0, 2]^d$, i.e., $[0, 2]^2$ for the visualized examples, see Fig. 10.3 for DBSCAN and Fig. 10.4 for k-means. Both figures compare the clustering results w.r.t. ratios $\varphi = 0.7$ and $\varphi = 0.3$ of selected solutions. DBSCAN uses the settings $\texttt{eps} = 0.3$ and $\texttt{min\_samples} = 5$, while k-means gets the known number of clusters, i.e. $k = 4$. The plots show that both algorithms are able to detect all clusters with exception of DBSCAN for ratio $\varphi = 0.7$ of selected solutions. This is due to the closeness of the four clusters, as the ratio of selected solutions is too large. Note, that k-means has the advantage of knowing the correct number of clusters.

In the following, we analyze the properties of the cluster result after clustering the exploratory solution set. Table 10.1 shows the number of correctly identified clusters, the intra- and inter-variance, and the DUNN index for $\lambda' = 1000$ and 10000 patterns, and ratios $\varphi = 0.1$ and 0.05 of selected solutions. The table shows that all parameter choices found the complete set of niches (4 out of 4). The inter-cluster variances

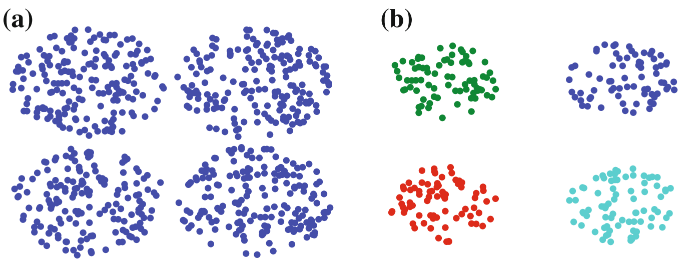

Fig. 10.3 Clustering results of DBSCAN (`eps` $= 0.3$ and `min_samples` $= 5$) on the niche benchmark problem for $\varphi = 0.7$ and 0.3 corresponding to $N = 700$ and $N = 300$. Patterns with same colors belong to the same clusters. **a** DBSCAN $\varphi = 0.7$. **b** DBSCAN $\varphi = 0.3$

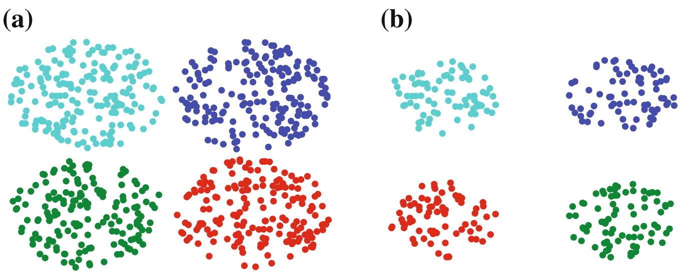

Fig. 10.4 Clustering results of k-means ($k = 4$) on the niche benchmark problem for $\varphi = 0.7$ and $\varphi = 0.3$. **a** k-means $\varphi = 0.7$. **b** k-means $\varphi = 0.3$

Table 10.1 Analysis of number of detected clusters, intra-, inter-cluster variance, and DUNN index for DBSCAN for various data set sizes N and ratios φ on the niche benchmark problem with $d = 2$

λ'	φ	#	Intra	Inter	DUNN
1000	0.1	4/4	0.0096	0.2315	1.6751
1000	0.05	4/4	0.0029	0.2526	3.7522
10000	0.1	4/4	0.0078	0.2517	1.8239
10000	0.05	4/4	0.0037	0.2508	3.0180

are larger than the intra-cluster variances. Further, the results show the intra-cluster variances shrink with higher selection ratio φ, as the patterns are missing that a further away from the niches' optima. This also results in a larger DUNN index value, as the diameters of the clusters are smaller and the clusters are further away from each other.

Now, we combine the explorative niche detection with the evolutionary optimization process employing a (1+1)-ES in each niche. After initialization of $\mathbf{x}$ with the cluster center that belongs to its niche, the evolutionary loop begins with the intra-cluster variance as initial step size σ. Figure 10.5 shows the optimization process of multiple (1+1)-ES with Rechenberg's step size control and $\tau = 0.5$. In each niche, an independent (1+1)-ES optimizes for 200 generations. The plots show the mean,

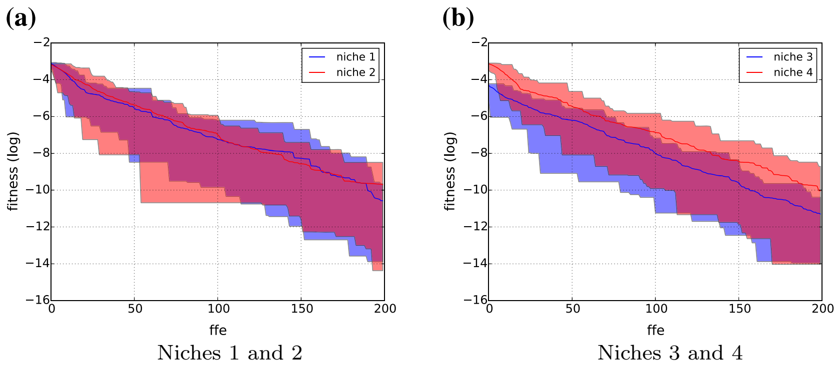

Fig. 10.5 Fitness development of 50 runs (mean, best, and worst runs) of four (1+1)-ES **a** in niches 1 and 2 and **b** in niches 3 and 4 running for 200 generations

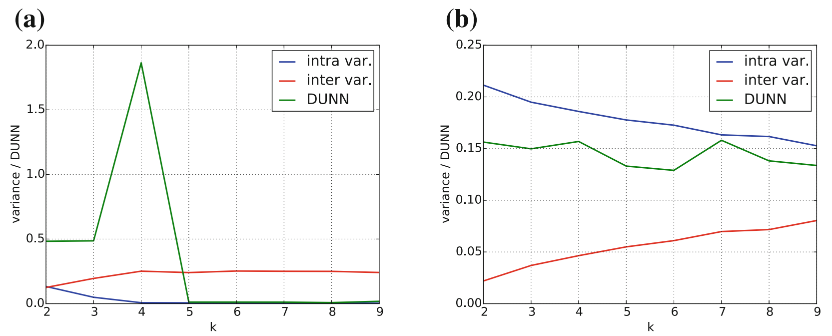

Fig. 10.6 Analysis of intra-cluster variance, inter-cluster variance, and DUNN index w.r.t. the number of clusters k when clustering with k-means **a** for $d = 2$ and **b** for $d = 10$

best, and worst fitness developments of 50 runs on a logarithmic scale. Our approach uses the mean of each cluster as initial solution and the square root of the intra-cluster variance as initial step size. The figures show that the optima are approximated logarithmically linear in each niche. An analysis of the approximated optima shows that the initial choice of step sizes is appropriate as no run of the (1+1)-ES leaves its assigned niche, and the logarithmically linear development starts from the early beginning.

Last, we analyze the dependency of the number of clusters when clustering with k-means on the three parameters intra-cluster variance, inter-cluster variance, and the DUNN index. Figure 10.6 shows the results when sampling with $\lambda = 10000$ points and rate $\varphi = 0.1$, i.e., $N = 1000$ for $d = 2$ on the left hand side and for $d = 10$ on the right hand side. The figures show that the inter-cluster variance increases with the number of clusters, while the intra-cluster variance decreases. In case of $d = 2$, a clear DUNN index maximum can be observed for $k = 4$. Due to the curse of dimensionality problem, the proper choice of cluster numbers does not show a similar DUNN index optimum for $d = 10$ like we observe for $d = 2$.

10.6 Conclusions

In multimodal optimization, the challenge is to detect multiple local optima. Niching is a technique that allows the detection and maintenance of multiple local and global optima at the same time during the evolutionary optimization process. This allows the practitioner to choose among alternative solutions if necessary.

In this chapter, we propose an approach of uniformly sampling patterns in solution space, fitness-based selection, and subsequent clustering for detection of potential niches. The basis for the detection of clusters is an exhaustive scan for basins in the initialization phase. This is performed with uniform random sampling in the allowed solution space. For the experimental test problem, we concentrate on the interval [0, 2] for each dimension. After the selection of the best solutions w.r.t. their fitness function values, clusters of potential basins appear. They are detected with k-means that affords the specification of the number of clusters before its run and DBSCAN that requires a definition of density parameters. The experimental analysis reveals that k-means and DBSCAN have proven to detect niches with proper parameter settings concerning the number of potential niches or solution space densities.

In the optimization phase, the approach to estimate step size σ with the intra-cluster variance is a successful concept to maintain the niches while optimizing with a (1+1)-ES. In these scenarios, we do not observe that a cluster is left or forgotten. The optimization in niches can naturally be parallelized and further blackbox optimizers can be applied.

References

1. Beyer, H., Schwefel, H.: Evolution strategies—a comprehensive introduction. Nat. Comput. **1**(1), 3–52 (2002)
2. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, (KDD 1996), pp. 226–231. AAAI Press (1996)
3. Dunn, J.: A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. J. Cybern. **3**(3), 32–57 (1973)
4. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Berlin (2003)
5. Shir, O.M., Bäck, T.: Niche radius adaptation in the CMA-ES niching algorithm. In: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature, PPSN IX 2006, pp. 142–151. Reykjavik, Iceland, 9–13 Sept 2006
6. Hansen, N., Ostermeier, A.: Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In: International Conference on Evolutionary Computation, pp. 312–317 (1996)
7. Pereira, M.W., Neto, G.S., Roisenberg, M.: A topological niching covariance matrix adaptation for multimodal optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2014, pp. 2562–2569. Beijing, China, 6–11 July 2014
8. Sadowski, K.L., Bosman, P.A.N., Thierens, D.: A clustering-based model-building EA for optimization problems with binary and real-valued variables. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015, pp. 911–918. Madrid, Spain, 11–15 July 2015

9. Preuss, M., Stoean, C., Stoean, R.: Niching foundations: basin identification on fixed-property generated landscapes. In: Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, pp. 837–844. Dublin, Ireland, 12–16 July 2011
10. Kramer, O., Danielsiek, H.: Dbscan-based multi-objective niching to approximate equivalent pareto-subsets. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2010, pp. 503–510. Portland, Oregon, USA, 7–11 July 2010
11. Bandaru, S., Deb, K.: A parameterless-niching-assisted bi-objective approach to multimodal optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2013, pp. 95–102. Cancun, Mexico, 20–23 June 2013
12. Biswas, S., Kundu, S., Das, S.: Inducing niching behavior in differential evolution through local information sharing. IEEE Trans. Evol. Comput. **19**(2), 246–263 (2015)
13. Hsu, P., Yu, T.: A niching scheme for edas to reduce spurious dependencies. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013, pp. 375–382. Amsterdam, The Netherlands, 6–10 July 2013
14. Cheng, S., Qin, Q., Wu, Z., Shi, Y., Zhang, Q.: Multimodal optimization using particle swarm optimization algorithms: CEC 2015 competition on single objective multi-niche optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015, pp. 1075–1082. Sendai, Japan, 25–28 May 2015,
15. Navarro, R., Murata, T., Falcon, R., Hae, K.C.: A generic niching framework for variable mesh optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2015, pp. 1994–2001. Sendai, Japan, 25–28 May 2015
16. Vargas, D.V., Takano, H., Murata, J.: Self organizing classifiers and niched fitness. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2013, pp. 1109–1116. Amsterdam, The Netherlands, 6–10 July 2013
17. Sheng, W., Chen, S., Fairhurst, M.C., Xiao, G., Mao, J.: Multilocal search and adaptive niching based memetic algorithm with a consensus criterion for data clustering. IEEE Trans. Evol. Comput. **18**(5), 721–741 (2014)