EA-Powered Basin Number Estimation by Means of Preservation and Exploration

Catalin Stoean[1], Mike Preuss[2], Ruxandra Stoean[1], and D. Dumitrescu[3]

[1] Department of Computer Science, Faculty of Mathematics and Computer Science,
University of Craiova, Romania
{catalin.stoean, ruxandra.stoean}@inf.ucv.ro

[2] Chair of Algorithm Engineering,
Department of Computer Science, Dortmund University of Technology, Germany
mike.preuss@uni-dortmund.de

[3] Department of Computer Science, Faculty of Mathematics and Computer Science,
Babes-Bolyai University of Cluj-Napoca, Romania
ddumitr@cs.ubbcluj.ro

Abstract. When using an evolutionary algorithm on an unknown problem, properties like the number of global/local optima must be guessed for properly picking an algorithm and its parameters. It is the aim of current paper to put forward an EA-based method for real-valued optimization to provide an estimate on the number of optima a function exhibits, or at least of the ones that are *in reach* for a certain algorithm configuration, at low cost. We compare against direct clustering methods applied to different stages of evolved populations; interestingly, there is a turning point (in evaluations) after which our method is clearly better, although for very low budgets, the clustering methods have advantages. Consequently, it is argued in favor of further hybridizations.

Keywords: Multimodal optimization, basins of attraction, function optimization, detect-multimodal mechanism

1 Introduction

The fitness landscape of an optimization problem that is considered for solving by means of evolutionary techniques is almost always completely unknown for the user. Exceptions are represented by the optimization of two- and three-dimensional functions that can be plotted in order to have an idea of the difficulty of the problem at hand. However, for the real-world tasks, one hopes for a unimodal problem, but usually expects that the landscape contains some local optima and one or more global ones. In this respect, it would be very useful to know in advance how multimodal the fitness landscape of the problem is, as this could help decide which optimization algorithm to choose or even set appropriate values for its specific parameters.

The aim of this paper is to design such a tool, also based on an evolutionary algorithm (EA), for the acquisition of data on the profile of the fitness landscape for problems defined over real-valued domains. Instead of obtaining a set of best solutions as usually pursued by contemporary niching EAs, we strive for obtaining an estimate on the number of optima an objective function possesses. One could imagine doing so by simply applying clustering techniques, but even these can only detect different clusters representing optima after somehow progressing towards good regions (as e.g. demonstrated in [1]); a random sample is hard to cluster meaningfully. In order to move into promising areas, some optimization method has to be applied before. However, marching too far e.g. by means of an EA implies the danger of missing several optima on which the subpopulations go extinct. Additionally, randomly initialized recombinative EAs have a natural tendency to contract the population near the search space center, as it has the lowest average distance to all individuals.

We must therefore find a good compromise between basin maintenance, convergence into basins, and further exploration. We track this goal by addressing the topology of the fitness landscape and the preservation of the fittest individuals, in a novel technique tailored after [2]. With this approach at hand, we compare against the straightforward clustering means – chosen as either the state-of-the-art Jarvis-Patrick or the more recent, effective Nearest-Better grouping – with a prior canonical EA for the generation of samples and a final unification of clusters based on the space topology. It goes without saying that the parametrization of any EA based method plays a decisive role for the ability to discover distinct optima and must be taken into account when fitting it for delivering estimates on the multimodality of unknown problems. Different parametrizations will influence the *reachable* search space region of the EA. Consequently, there are no means to perform estimations over areas never visited.

The paper is organized as follows. §2 emphasizes the circumstances and the arguments for the development of such an instrument, while §3 describes the Topological Multimodality Estimator (TME). Conducted experiments to validate and investigate the estimations of proposed technique are outlined in §4; the two clustering algorithms also examine test landscapes and results of expected/found optima of all three are put side by side. Finally, conclusions of the experimentation and outcome are reached.

2 Context and Motivation

When a less-known multimodal problem is considered, one may either resort to *iterated local search* techniques (ILS) [3] or rely on a usually radius-dependent *niching* EA for separating the resembling individuals into different subpopulations (species); in the best case, each one of these would track a different optimum and the number of subpopulations equals the number of optimal solutions. When such an EA is employed, the main concern lies in determining an accurate value for the radius parameter that would help separating the individuals into subpopulations in the most advantageous manner.

Among the radius-related EAs, the most commonly referred is the niching technique of Goldberg and Richardson [4]; for the last two decades, it has represented a source of inspiration for the development of many radius-based EAs in the vein of it [5], [6]. Within this niching technique, individuals are grouped into species by a given radius, so that no distance inbetween them is larger. As previously stated, the value to be chosen for the radius parameter directly depends on the fitness landscape, i.e. on the problem considered for solving. Selecting an appropriate value for the radius assures accurate results. Deb and Goldberg [7] proposed one solution for computing the value for the radius threshold (σ_{share}) that leads to the formation of subpopulations; this has afterwards been embraced by most of the researchers dealing with such parameters. Knowing the number of optima that are to be found, N, and being aware that each niche is enclosed by an n-dimensional hypersphere of radius r, the niche radius σ_{share} can be estimated as $\sigma_{share} = \frac{r}{\sqrt[n]{N}}$.

However, in most of the cases, especially for real-world applications, one usually cannot know in advance the number of optima. Additionally, there is no guarantee that basins of attraction are formed like regular hyperspheres. Methods for approximating the number of solutions for combinatorial optimization problems are described e.g. in [8]. For problems with continuous domain, by investing a small amount of fitness evaluations and using the tool that we put forward, one could have an approximation for N, or at least for the fraction of N that is found relatively often. We assume that this *reachable* fraction of optima heavily depends on the configuration of the underlying EA, so that it is larger for more explorative settings. For any clever technique, a higher number of fitness evaluations invested shall lead to a more accurate estimation, as is the case for the proposed method. Moreover, the present approach also provides approximations of the detected optima, especially for relatively high budgets of evaluations.

3 Topological Multimodality Estimator

As the suggested method represents a pre-processing tool, it shall provide information at a very low cost, i.e. with a reduced budget of fitness evaluations. In order to achieve this aim and, at the same time, explore the search space thoroughly, we utilize a variable sized population: We start with a large population and subsequently continue solely with the most prolific individuals that belong to different basins of attraction. Thus, the number of consumed fitness evaluations is kept low. The population size is allowed to raise again during reproduction, but, unless new basins of attraction are discovered, it is again reduced to a minimum. Another important constraint that had been taken into account is that the technique did not have to require additional parameters (as compared to a canonical EA) that directly depend on the considered problem. In order to avoid the use of a threshold (radius) for subpopulation differentiation, we exploit the topology of the fitness landscape, separating us from simple clustering approaches.

The algorithm begins with the generation of random individuals within the problem domain. A method for detecting whether two individuals belong to the same basin of attraction or not is succeedingly used for selecting the fittest individuals within each of the detected attractors. The procedure was introduced in [9] and called *hill-valley*; taking into account two individuals, it verifies whether there exists either a *hill* (in this case, they track the same peak) or a *valley* (different peaks) inbetween, within the fitness landscape. The method is herein renamed *detect-multimodal* for reasons of simplicity and is described below. From this point on, the search continues only with the fittest individuals that undergo recombination and mutation; obtained offspring are checked to see whether they belong to different basins of attraction than the ones already discovered and the population is updated by retaining the best individual within each attractor. Evolution continues for several cycles until the predefined budget is consumed.

3.1 The Detect-Multimodal Mechanism

The routine takes two individuals (points) as input, checks their relative position within the search space and returns a boolean value, which specifies whether there is a valley between them in the fitness landscape or not: In the latter circumstances, the conclusion is that they climb different hills. In order to reach that decision, a set of interior points between the twois generated. If the fitness of all these is higher than the minimal fitness of the two tested individuals, it is concluded that they track the same optimum. Contrarily, if there exist such a point whose fitness is smaller than the minimal fitness of the two, then it is assessed that they follow different peaks. To conclude, the *detect-multimodal* method verifies the assumption that two individuals track the different optima and returns true if so and false if they follow the same peak [9]. The only required parameter refers to the number of gradations (interior points) taken into account. In all undertaken experiments of current paper, the gradations are values taken equidistantly from the interval [0,1]. The higher the number of interior points, the more precision the outcome of the *detect-multimodal* mechanism has.

3.2 TME Mechanics

TME starts with the initialization of a uniformly randomly generated set of individuals. From this collection, the fittest individual from each different basin of attraction is selected. The chosen individuals undergo an iterative process that includes the following steps. Recombination is applied to the selected pool of individuals: All offspring obtained after recombination are added to the current population which is subject to mutation. The membership of all offspring to the currently detected basins of attraction is verified. For each discovered basin, only the fittest individual is kept. The individuals located in previously unseen basins are also preserved, once more only the fittest one per basin. The selection of the fittest individual within every attraction basin uses the *detect-multimodal* procedure for distinguishing the different attractors (Algorithm 1). The entire population is sorted decreasingly according to fitness. The fittest individual in the population represents a seed. Each individual in the sorted series is considered in turn and checked against the currently found seeds to see if they track distinct optima. If it follows a different peak than all the others that have been tracked until the present moment, then the individual represents a new seed. The recorded seeds (*Seeds*) are taken

Algorithm 1 Seeds identification

```
Sort population $P$ decreasingly according to fitness;
$Seeds$ = $\{P_1\}$; //the fittest individual automatically becomes a seed
for $i$ = 2 to $n$ do
    Find closest (highest probable) $s \in Seeds$ such that detect-multimodal($P_i$, $s$) = $false$;
    if no such a seed $s$ then
        $Seeds = Seeds \cup \{P_i\}$; // $P_i$ is a seed
    end if
end for
return the $Seeds$ set
```

Algorithm 2 Integration of the newly created individuals

```
for each offspring $x$ in $X$ do
    Find its closest individual $s$ in $Seeds$ for which $detect - multimodal(x, s) = false$;
    if $s$ exists then
        $Seeds = (Seeds \setminus \{s\}) \cup \{fitter(x, s)\}$; // $x$ and $s$ fight for survival
        $X = X \setminus \{x\}$;
    end if
end for
Find the fittest free individual $x$ in $X$;
$NewSeeds = \{x\}$; // $x$ is a new seed
while there are still individuals in $X$ do
    For fittest $x \in$ X find closest $s \in NewSeeds$ such that $detect-multimodal(x, s) = false$;
    if no such seed $s$ then
        $NewSeeds = NewSeeds \cup \{x\}$;
    end if
    $X = X \setminus \{x\}$;
end while
$Seeds = Seeds \cup NewSeeds$; // individuals that follow other peaks are added to population
return the $Seeds$ set with the integrated individuals
```

one by one in terms of Euclidean proximity from the considered individual. It is more likely that the individual tracks the same peak as the nearest seed and, consequently, it is verified, by distance rank, against the closest ones to avoid unnecessary calls of the *detect-multimodal* procedure.

The selected set of seeds then enters the evolutionary cycle. Thus, the size of the population is drastically diminished in order to reduce the fitness evaluation cost. Recombination takes place and all resulted offspring are appended to the current population. Hence, the space between the currently tracked optima is explored. Now the whole population undergoes mutation. All obtained offspring from either of the two variation operators are checked against the parent population with two purposes. If an offspring tracks an optimum that has already been followed by one closest individual from the parent population, i.e. the two belong to the same basin of attraction, then only the fitter of the two is kept in the seeds population for the next generation. Secondly, when an offspring lies within a basin of attraction that has not been previously tracked by any other individual from the parent population, it shall be added to the seeds population of

Algorithm 3 Topological Multimodality Estimator

```
Initialize population and identify the seeds $Seeds$ (Algorithm 1);
while stop condition is not met do
    Apply mating selection to $Seeds$;
    Apply recombination to $Seeds$ and obtain the set of offspring $X$;
    Apply mutation to $Seeds$ and $X$ and append all obtained offspring to $X$;
    Integrate the newly created individuals X to $Seeds$ (Algorithm 2);
end while
return the cardinal of $Seeds$ and the actual solutions in $Seeds$
```

the next generation, given that there are no solutions between itself and other descendants that lie within the same basin of attraction (Algorithm 2). *NewSeeds* represents the set of seeds that are detected in the current generation. Each time a new seed is considered for adding, it is checked against the other solutions in the *NewSeeds* set. Finally, *NewSeeds* is appended to the *Seeds* set to form the population that will enter the next generation. The steps of the entire approach are outlined in Algorithm 3. Note that with TME, basin identification in the worst case requires $O(|P|^2)$ extra evaluations ($|P|$ stands for population size). However, this happens only if every newly produced individual is outside all yet identified basins. Additionally, the population is kept small by deleting all non-seed individuals inside a basin. Practical experience shows that with these measures, the process requires rather $O(|P|)$ evaluations for basin testing.

4 Experimental Investigation

Experimentation aims to validate the proposed technique against functions whose number and location of optima is known and compare the performance to that of the cheap alternative of direct clustering on the search space. The Waves function ($F1$, 10 optima) is asymmetric and has some peaks difficult to find as they lie on the border or on flat hills. The Six-Hump Camel Back function ($F2$, 6 optima) exhibits two local optima that are not really higher than their neighboring regions and thus can easily be missed. Additionally, we employ a highly multimodal function (Rastrigin, F3) as model for problems for which neither location nor number of optima is known (it is clear that these are easy to compute in this case). Here, the global optimum is surrounded by a large number of close local optima with small relative differences in their values.

$$\mathrm{F1}(x,y) = (0.3x)^3 - (y^2 - 4.5y^2)xy - 4.7cos(3x - y^2(2+x))sin(2.5\Pi x)),$$
$$-0.9 \leq x \leq 1.2, -1.2 \leq y \leq 1.2$$
$$\mathrm{F2}(x,y) = -((4 - 2.1x^2 + \tfrac{x^4}{3})x^2 + xy + (-4 + 4y^2)y^2),$$
$$-1.9 \leq x \leq 1.9, -1.1 \leq y \leq 1.1$$
$$\mathrm{F3}(x) = -(10 \cdot n + \textstyle\sum_{i=1}^{n}(x_i^2 - 10 * cos(2 \cdot \Pi \cdot x_i))), -5.12 \leq x_i \leq 5.12.$$

Two conceptually different nearest neighbor clustering approaches were taken into account for comparison: The Jarvis-Patrick clustering method and the recently proposed Nearest-Better algorithm [10]. A canonical EA evolves a population of individuals for a number of fitness evaluations and clustering is subsequently applied to the final generation. The estimated number of basins is given by the resulting number of clusters, while the approximate optima are given by the prototypes. The Jarvis-Patrick (JP) algorithm [11] considers a list of J nearest neighbors—in terms of (Euclidean) distance—for each individual. Every point in the search space is verified in turn against all others: 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. A point cannot belong to more than one cluster. Moreover, if x and y meet the condition to belong to the same cluster and x and z also pass the two criteria, all three will be clustered together, indifferent of the fact of whether y and z also respect the conditions. Finally, the prototypes are determined as the fittest individuals in each cluster. The drawback of this very efficient algorithm consists in the two parameters J and K that results are very dependent upon. The recently introduced Nearest-Better (NB) clustering mechanism relies on the connection to one immediate neighbor for each point, which is also better in terms of fitness—thus, topological information is included in addition to location of points. It 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. Every individual connects to its nearest better neighbor (in terms of Euclidean distance once more). The longest edges—those higher 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. This approach possesses only one additional parameter to be tuned, with 2 being a good default value according to [10].

Research Questions: How do TME and JP/NB compare in terms of performance on functions with known number of solutions? Can we find a correlation of results obtained for the same configuration but in different runs to estimate the 'reachable' optima?

Pre-experimental Planning: The two selected clustering techniques are utilized on the test cases in advance in order to get acquainted to their behavior. The preceding canonical EA also stops after a fixed number of fitness evaluations. Comparing the number of detected optima against the number of clusters lead to the insight that the two methods largely overestimate the number of attraction basins for both functions, with an advantage on the NB side which is less deceiving. The number of clusters was approx. 3 times higher than the amount of optima. The overrating clustering action of the two techniques had to be resolved, in order to set an equal rival to suggested TME. Ergo, we applied the *detect-multimodal* mechanism with a limited number of interior points (set to 2 in the undertaken experiments) after clusters are determined, in order to unify groups within the same basin. The fitness evaluations employed in this final step are also counted within the totally allowed value.

Task: Directly compare the number of attraction basins found (F1 and F2) or reachable basins estimated (F3) by TME and JP/NB clustering. Measure the ability of the techniques to find the same solutions in multiple runs of the same parameter design.

Experimental Setup: The same budget of fitness evaluations was used both for TME and JP/NB, ranging from 200 to the maximum 2000. The values for all parameters were generated using a Latin Hypersquare Design, i.e. 30 space-filling configurations were produced. The parameters of the evolutionary algorithm were generated within the following intervals for all three methods: Population size is between 2 and 200, mutation and recombination probabilities between 0 and 1 and mutation strength between 0 and 5. Additionally for the JP method, the values for the two parameters J and K were both created between 1 and 25 with the constraint that $J > K$. Plus, as the number of neighbors cannot be higher than the population size, the latter is between 25 and 200. The TME technique also makes use of one parameter, which is the number of interior points considered for the *detect-multimodal* method. The positive integer is generated between 1 and 15. In order to evaluate whether a technique tends to find the same optima in different runs with the same parameter configuration (estimation of *reachable* optima number), we conducted the following computations: Within each parameter design, for every two runs (out of 30 performed here), we use the pair of solution sets A and B for computing the fraction of commonly found (correlated) peaks $d = \frac{|A \cap B|}{|A|}$, where |A|

represents the cardinal of the set A. Averaging $d/\mid B \mid$ leads us to an estimator for $1/n$ and thus for n according to [12]. However, we assume that here n is not the number of all optima but rather the number of *reachable* basins for each different configuration. Note that although used with results from 30 runs here, the estimator shall get stable already for very small run sets. Otherwise, it would suffice to simply count the number of optima found within a large number of runs.

Results/Visualization: Table 1 gives the number of optima detected by the three compared techniques on F1 and F2, for the (30 LHD) different algorithm parameter settings. $Best$ columns refer to the highest average number of optima out of all configurations, whether $average$ stands for the average of all runs of all configurations. The small differences between best and average TME results prove the fact that it is not very sensitive to the parameter values.

Table 1: *Attraction basins found by TME, JP and NB in the best configuration and average over 30 configurations for $F1$ and $F2$ with different fitness evaluation budgets.*

Fitness	F1						F2					
evaluation	TME		JP		NB		TME		JP		NB	
calls	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average	Best	Average
200	5.96	4.48	8.13	5.15	8.36	5.95	3.4	2.97	4	3.33	4	3.37
500	6.96	5.68	8.2	4.33	8.26	5.16	4.43	3.63	4	2.62	4	2.57
1000	8.1	6.71	8.1	2.68	8.56	3.24	5.16	4.08	3.96	1.75	3.83	1.7
2000	9.33	7.89	2	1.07	2.5	1.21	5.63	4.45	1.93	1.22	1.96	1.24

As for the second part of the experiment, which regards the correlation of basin sets, when JP and NB were applied for $F3$ with 5 or more variables, it was found that for several configurations, in all the 30 repeats, the methods found only different attraction basins. In such a case, the value for the estimation n is infinity, which is not a meaningful a priori information about the problem landscape. Consequently, the results of JP and NB for $F3$ with 5 or 10 variables are not reported. For $F1$ with 200 evaluation calls, NB provides the highest averaged value over all configurations for n, 6.08, while for TME this is 5.34 and for JP 5.14. The maximum value in one configuration in these low budget conditions is obtained by NB (8.32). As the number of evaluation calls is increased, the average value for n puts TME in advantage and lowers the values for the clustering methods: For 2000 evaluation calls, the average n for TME is 7.98, while for JP and NB they are 1.26 and 1.4, respectively. The situation is very similar for $F2$, where 3.42 is the estimated reachable basin number for NB, 3.28 when JP is employed, and 3.08 for TME when the lowest budget is used; the value moves up to 4.42 for the highest budget considered for TME, while it goes down for JP and NB towards 1.92 and 1.88. For $F3$ (2 var.), TME has a larger set of solutions that are found in multiple runs, even with 200 evaluations calls: TME has 16.62, NB gives 14.62, while JP has 12.23. The value increases again for TME up to 20.01 for the highest budget, while NB has 3.74 and JP 2.17, in the same circumstances. In case of 5 and 10 variables, TME estimates in average 43.42 and 40.21 solutions (highest budget) and drops to 24.2 (5 var.) and to 17.14 (10 var.), respectively (lowest budget). The configurations with the highest values find n equal to 90.5 (5 var.) and 71.2 (10 var.), while the smallest value in one configuration is around 9 for both cases.

Observations: While JP and NB perform very well for a small number of fitness evaluations and tend to decrease quality as more are considered, TME goes in the opposite direction as a higher budget assures significantly better results. JP and NB with the subsequent *detect-multimodal* manage to overcome the initial overrating. The explanation for the performance decrease in JP and NB for increased number of evaluations lies in the fact that the optimization process drives the population towards one or few optima and, therefore, other local optima are neglected as they remain empty. For both functions $F1$ (10 optima) and $F2$ (6 optima), the closest to the correct solution is TME, when the 2000 evaluation calls are considered. It is interesting to see that JP and NB come very close the real number of basins when the lowest budget is considered (8.36 out of 10 for $F1$ and 4 out of 6 for $F2$). However, it shall also be noted that the difference from the best configuration to the average over all configurations is very high in all cases for the clustering techniques, while there exists only a small such difference for TME. It seems that TME is not very dependent on the values of its parameters, while for JP and NB they play an important role as *wrong* configurations lead to poor results.

Discussion: While JP and NB perform better for lower evaluation costs, TME's accuracy is significantly increased when the evolutionary cycle is prolonged. This is of course due to the interaction between the basin preservation and detection and explorative phases in TME. In JP and NB, this is inexistent and hence, the underlying EA produces smaller and smaller basin numbers while the runs progress. Out of the objective comparison intention of this experiment, further tests were undertaken for TME with a budget of up to 6000. The solutions quality was gradually increased until it reached 9.9 attraction basins for $F1$ and all 6 for $F2$ for the best configuration, in average over 30 repeats. The two presented clustering methods, JP and NB, represent good economical alternatives for estimating the number of different attraction basins of a fitness landscape. However, they strongly depend on the underlying optimization algorithm, so that providing more evaluations does not result in improvements unless the underlying algorithm itself is explorative *and* preserving. However, if such means are provided, as in TME, one gets a much less parameter-dependent and thus robust method, which shall be the better choice, especially for real-world problems. Nevertheless, for higher dimensional spaces, results attest an increased level of parameter-dependency; additional investigations are necessary to observe what parameter settings make TME efficient.

5 Conclusions and Future Work

An evolutionary technique, Topological Multimodality Estimator, to determine the profile of the fitness landscape for real-valued optimization problems, with respect to a low budget of evaluation calls, is introduced. A variable sized population to keep only the most promising solutions for further evolution is thus used. Two improved clustering methods, applied to the set of solutions of a canonical EA, are considered for direct comparison on three multimodal functions. For two dimensions and a very low number of evaluation calls, clustering provided better results. However, when the techniques are allowed to evolve for more generations, TME results improve, while the quality of the EA/clustering method combinations is worse – the underlying EA converges to few attractive regions as it has no means to preserve the already found basins.

Rethinking the obtained results, the conclusion seems obvious. *Simple* clustering methods are cheap and successful especially during the early phases of an optimization, and explorative methods like TME need more evaluations to obtain comparable results, but do have a much higher potential; investing more yields more. Hybridization of TME with one of the clustering methods to eliminate valuable calls of the fitness function, especially at the beginning of the evolutionary cycle, may be a middle alternative. At the same time, clustering affects the quality of solutions when the evaluation calls budget is increased; this would be overcome through the replacement of the canonical EA by the TME engine. Out of the two, NB looks like a better alternative for hybridization, not only because of the better results, but also for the absence of additional parameters. A future TME version should also attain information on the sizes of the detected attraction basins and, finally, the influence of its estimated number of optima on the success probability/convergence rate of a radius-based search algorithm will be investigated.

References

1. Parmee, I. C.: The Maintenance of Search Diversity for Effective Design Space Decomposition Using Cluster-Oriented Genetic Algorithms (COGA) and Multi-Agent Strategies (GAANT). In: 2nd International Conference on Adaptive Computing in Engineering Design and Control (ACEDC '96), pp. 128–138. University of Plymouth (1996)
2. Stoean, C., Preuss, M., Stoean, R., Dumitrescu, D.: Disburdening the Species Conservation Evolutionary Algorithm of Arguing with Radii. In: Lipson, H. (ed.) 9th Annual Conference on Genetic and Evolutionary Computation (GECCO '07), pp. 1420–1427. ACM Press (2007)
3. Ramalhino Lourenco, H., Martin, O., Stützle, T.: Iterated Local Search. In: Glover, F., Kochenberger, G. A. (eds.) Handbook of Metaheuristics. Kluwer (2002)
4. Goldberg, D. E., Richardson, J.: Genetic Algorithms with Sharing for Multimodal Function Optimization. In: 2nd International Conference on Genetic Algorithms, pp. 41–49. Lawrence Erlbaum Associates, Hillsdale, NJ (1987)
5. Mahfound, S. W.: Niching Methods for Genetic Algorithms. Technical Report, IlliGAL, 95001, University of Illinois (1995)
6. Beasley, D., Bull, D. R., Martin, R. R.: A Sequential Niche Technique for Multimodal Function Optimisation. J. Evol. Comp., vol. 1, no. 2, pp. 101–125 (1993)
7. Deb, K., Goldberg, D. E.: An Investigation of Niche and Species Formation in Genetic Function Optimization. In: 3rd International Conference on Genetic Algorithms, pp. 42–50. Morgan Kaufman, San Mateo, California (1989)
8. Reeves, C. R., Eremeev, A. V.: Statistical Analysis of Local Search Landscapes. JORS, vol. 55, no. 7, pp. 687–693 (2004)
9. Ursem, R. K.: Multinational Evolutionary Algorithms. In: Congress of Evolutionary Computation (CEC '99), vol. 3, pp. 1633–1640. IEEE Press, Piscataway, NJ, (1999)
10. Preuss, M., Schoenemann, L., Emmerich, M.: Counteracting Genetic Drift and Disruptive Recombination in $(\mu \overset{+}{,} \alpha)$-EA on Multimodal Fitness Landscapes. In: Beyer, H. G., O'Reilly, U. M. (eds.) 7th Annual Conference on Genetic and Evolutionary Computation (GECCO '05), pp. 865-872. ACM Press (2005)
11. Jarvis, R. A., Patrick, E. A.: Clustering Using a Similarity Measure Based on Shared Near Neighbours. J. IEEE Trans. on Comp., pp. 1025-1034 (1973)
12. Jelasity, M., Mike Preuss, M.: On Obtaining Global Information in a Peer-to-Peer Fully Distributed Environment. Euro-Par'02, pp. 573-577 (2002)