

Distance-Based Tournament Selection

Christian Oesch^(✉)

Faculty of Business and Economics, University of Basel, Basel, Switzerland
christian.oesch@unibas.ch

Abstract. In this paper we analyze the performance of a novel genetic selection mechanism based on the classic tournament selection. This method tries to utilize the information present in the solution space of individuals, before mapping their solutions to a fitness measure. This allows to favour individuals dependent on what state the evolutionary search is in. If a population is caught up in several local optima, the correlation of the distance between the individuals and their performance tends to be lower than when the population converges to a single global optimum. We utilize this information by structuring the tournaments in a way favorable to each situation. The results of the experiments suggest that this new selection method is beneficial.

Keywords: Genetic algorithms · Tournament selection · Search space

1 Introduction

Multiple optima are still a challenge in many optimization problems. With standard approaches, one always balances exploitation of locality against exploration of the search space. In genetic algorithms [3] this is done by choosing the right evolutionary parameters and genetic operators. However, these values are fixed over the course of the optimization and do not take into account the current state of the population or where it is located in the search space. Genetic algorithms (GAs) base the guidance of their search on a single measure; the fitness. This single-value metric defines which individuals reproduce, and, over several generations, in which areas of the search space the population concentrates. In this study we want to open the discussion on using the genetic distribution of the population to guide the search process. To do this, we propose the use of genetic distances between individuals to modify the tournament composition in the selection stage of genetic algorithms. By measuring the correlation of the distances between individuals and their fitness, we believe that we can introduce mechanics which will enhance the algorithms performance.

This approach is related to niching methods [7, 8] but does not rely on fixed distances and is adaptive to the state of the convergence process. It is an example of how to use additional information to guide the search process. In our method, depending on an individual's place in the population, only the chance of competing in a given tournament is modified. This results in a softer modification of the standard approach.

The rest of this article is structured as follows. In Sect. 2 the proposed method is presented. Section 3 describes the test cases and the evolutionary settings employed in this study. The results of the experiments are presented in Sects. 4 and 5 concludes this study and introduces further ideas and future areas of research.

2 Methodology

There are different ways to influence a GA's convergence behavior through additional information. In this study, we influence the convergence of the algorithm by modifying which opponents compete in a tournament. Note that this is an alternative approach to fitness sharing [12], fitness penalties [5] or other measures which directly change the selection probability of an individual. Instead, depending on the state of the genetic algorithm, we alter a tournament's opponents which can be genetically more or less diverse. In case of a very homogeneous tournament, meaning the individuals are similar, there is a high chance that the best individual of the local cluster will be selected. In more heterogeneous tournaments, as the individuals will not always have to compete against the best individual in the cluster, individuals in more distant regions of the search space have a higher chance of selection. Instead of modifying directly which individuals are selected, we basically give certain individuals a higher chance of participating in a tournament.

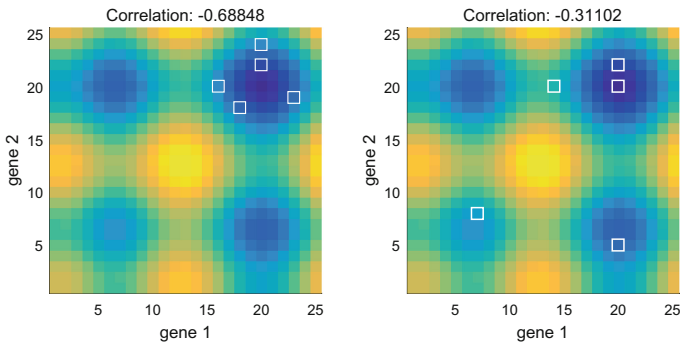


Fig. 1. The correlation of average distances to the other individuals and the individual's fitness. The dark blue spot around (20, 20) is the optimum. When the individuals gather in one optimum, the negative correlation of fitness to the average distance to other individuals tends to be stronger than in the case where some individuals are located in a local optima. (Color figure online)

Figure 1 shows an example for the two different cases of a negative correlation between the individual's average distance to each other and their fitness. In the left panel, the individuals gather in one optimum and the most central

individual¹ of the population exhibits the highest fitness. On the right hand side, the correlation is weaker, as the best individual is no longer in the spatial center of the population. The optimal tournament will look different in both cases.

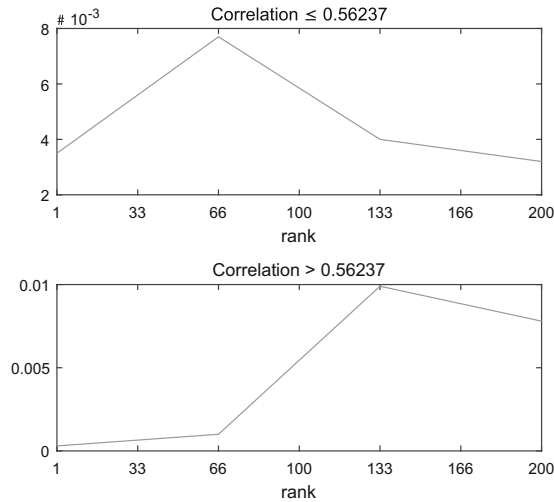


Fig. 2. Two probability transformations based on the rank distance to the initial tournament candidate. The x-axis shows the rank distance, the y-axis shows the probability of this rank's selection into the tournament.

In each tournament, the initial candidate is chosen at random. This is done uniformly and does not differ from standard tournament selection. However, once the initial candidate is chosen, the other constituents of the tournament are chosen based on their rank distance. To build the rank-distances between the individuals, we initially measure the Canberra-Distance [6] of their genomes. The Canberra-Distance is used such that the different dimensions have similar importance. It is defined by:

$$d(u, v) = \sum_{i=1}^n \frac{|u_i - v_i|}{|u_i| + |v_i|}, \quad (1)$$

where u and v are the vectors to compare, in this case the genomes. These distances are then turned into ranks r from 1 to N where N is the population size. Depending of the state of the algorithm a probability transformation is chosen to either favor individuals which are close to the initial candidate or solutions that are genetically further apart. This transformation is then applied to the rank distances to the other individuals and for each individual of the population a probability is assigned with which that individual is chosen for the current spot in the tournament. The probability of individual j to end up in a particular tournament p_j is then:

¹ The one with the lowest average distance to other individuals.

$$p_j = \begin{cases} g_1(r_j) & \text{if } \rho \geq \alpha \\ g_2(r_j) & \text{if } \rho < \alpha \end{cases} \quad (2)$$

where ρ is this generation's correlation between the average distance between individuals to their fitness score, and alpha is a threshold. The selection into the tournament is then done akin to a roulette wheel selection. This process is repeated for all opponents in the tournament. Once all opponents have been found, the tournament is performed and the tournament winner is determined and marked for reproduction and survival.

Figure 2 shows two probability transformations to select further individuals into the tournament. The x-axis shows the rank distance to the initial tournament candidate, and the y-axis the probability of the corresponding individual to be selected into the tournament. The transformations have been determined by Monte Carlo search over the Schwefel 1.2 problem ($f_2(x)$), shown in Sect. 3. This problem has been chosen as over the course of the runs, it has a wide range of values of ρ .

3 Experiments

To test the new concept, we apply it to most of the benchmark problems proposed in [14]. The functions are listed in Table 1. Functions f_1 to f_7 are unimodal, and the functions f_8 to f_{14} multimodal. The parameter n defines the dimensionality of the problem and S the range. f_{\min} shows the minimum value of the optimization problem.

The functions are well-known benchmarks from the literature. $f_1(x)$ is a simple sphere model, $f_2(x)$, $f_3(x)$, $f_4(x)$ are Schwefel's problems 2.22, 1.2, and 2.21 [13]. $f_5(x)$ is the generalized Rosenbrock's function [11]. $f_6(x)$ is a step function and $f_7(x)$ is a quartic function with noise. A generalized version of Schwefel's problem 2.26 is given in $f_8(x)$. $f_9(x)$ is the generalized Rastrigin's function [10], and $f_{10}(x)$ Ackley's function [1]. The benchmark function $f_{11}(x)$ is the generalized Griewank function [4], and $f_{12}(x)$ Shekel's Foxholes function [9]. $f_{13}(x)$, f_{14} , and $f_{15}(x)$ are Kowalik's [14], Six-Hump Camel-Back [9], and Branin [2] functions.

All experiments are conducted with the same evolutionary parameters listed in Table 2. 300 runs are performed for both the standard tournament selection as well as for the novel distance-based tournament. The fitness is calculated by the mean-squared error (MSE). As noted in the previous section, problem $f_2(x)$ has been used to find a piecewise-linear probability transformation by Monte Carlo search. The parameters of the piecewise function are listed in Table 3. We apply two different probability transformations $g_1(r)$ and $g_2(r)$ depending on ρ . The ranks are split into three different segments, and the probabilities of selection into the tournament are interpolated based on the rank of the individuals.

Table 1. Benchmark functions

| Test function | n | S | f_{\min} |
|--|----|---------------------------|------------|
| $f_1(x) = \sum_{i=1}^n x_i^2$ | 30 | $[-100, 100]^n$ | 0 |
| $f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $ | 30 | $[-10, 10]^n$ | 0 |
| $f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$ | 30 | $[-100, 100]^n$ | 0 |
| $f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$ | 30 | $[-100, 100]^n$ | 0 |
| $f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30 | $[-30, 30]^n$ | 0 |
| $f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$ | 30 | $[-100, 100]^n$ | 0 |
| $f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$ | 30 | $[-1.28, 1.28]^n$ | 0 |
| $f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$ | 30 | $[-500, 500]^n$ | -12569.5 |
| $f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | 30 | $[-5.12, 5.12]^n$ | 0 |
| $f_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$ | 30 | $[-32, 32]^n$ | 0 |
| $f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 30 | $[-600, 600]^n$ | 0 |
| $f_{12}(x) = [\frac{1}{500} + \sum_{j=1}^2 5] \frac{1}{\prod_{i=1}^n (x_i - a_{ij})^6}$ $a_{ij} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & 32 & 32 & 32 \end{bmatrix}$ | 2 | $[-65.536, 65.536]^n$ | 1 |
| $f_{13}(x) = \sum_{i=1}^4 1[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]$ $a_i = (0.1957, 0.1947, 0.1735, 0.1600, 0.0844, 0.0627, 0.456, 0.0342, 0.0323, 0.0235, 0.0246)$ $b_i^{-1} = (0.25, 0.5, 1, 2, 4, 6, 8, 10, 12, 14, 16)$ | 4 | $[-5, 5]^n$ | 0.0003075 |
| $f_{14}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_i^6 + x_1x_2 - 4x_2^2 + 4x_2^4$ | 2 | $[-5, 5]^n$ | -1.0316285 |
| $f_{15}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8}\pi) \cos x_1 + 10$ | 2 | $[-5, 10] \times [0, 15]$ | 0.398 |

4 Results

Table 4 shows the percentage of the mean-squared error in the distance-based tournament compared to the baseline runs for different quantiles and the mean. For example for problem $f_1(x)$ the minimum MSE at the end of the run was 64.4% of the baseline result. It is clear to see that our method dominates the standard implementation in most of the problems. Except for $f_{14}(x)$ and $f_{15}(x)$ the median of the MSE is smaller in the distance-based approach. Especially in the unimodal case, the new structure of the tournament performs well. Remember that in these cases, the correlation between the average distance to the other individuals and the fitness is most likely high. This means, the probability transformation $p_1(r)$ is applied predominantly. The method also outperforms the standard approach in the multimodal functions. However, the difference is not as large. Further performance improvements might be achieved by performing a more thorough search for the parameters of g_1 and g_2 . The results are further presented in Fig. 3. The figure shows box plots of the results of all the benchmark problems. The reason for why the method does not work as well for functions

Table 2. Settings

| Parameter | Value |
|-----------------------|------------|
| Number of runs | 300 |
| Generations | 400 |
| Population size | 200 |
| Crossover probability | 0.8 |
| Mutation probability | 0.065 |
| Selection | Tournament |
| Tournament Size | 8 |
| Fitness | MSE |

Table 3. Parameters of the probability transformation

| | $p(r_{0.0})$ | $p(r_{0.33})$ | $p(r_{0.66})$ | $p(r_{1.0})$ |
|-------------------------|--------------|---------------|---------------|--------------|
| $g_1 : \rho > 0.562$ | 0.0035 | 0.0077 | 0.0040 | 0.0032 |
| $g_2 : \rho \leq 0.562$ | 0.0003 | 0.0010 | 0.0099 | 0.0078 |

Table 4. Percentage of baseline MSE

| | Min | $Q_{.25}$ | $Q_{.5}$ | $Q_{.75}$ | Max | Mean |
|-------------|---------|-----------|----------|-----------|--------|--------|
| $f_1(x)$ | 64.4% | 80.2% | 79.9% | 80.7% | 75.3% | 79.7% |
| $f_2(x)$ | 71.1% | 91.4% | 91.4% | 92.2% | 98.0% | 91.9% |
| $f_3(x)$ | 68.6% | 83.3% | 81.4% | 83.4% | 66.9% | 81.0% |
| $f_4(x)$ | 112.5% | 92.3% | 93.1% | 93.9% | 96.7% | 93.6% |
| $f_5(x)$ | 75.0% | 90.3% | 89.9% | 82.6% | 107.6% | 81.5% |
| $f_6(x)$ | 106.0% | 83.4% | 85.1% | 87.5% | 92.2% | 84.5% |
| $f_7(x)$ | 99.5% | 99.0% | 98.8% | 99.8% | 100.0% | 99.1% |
| $f_8(x)$ | 108.8% | 83.9% | 83.4% | 82.6% | 93.2% | 84.1% |
| $f_9(x)$ | 89.8% | 90.4% | 91.5% | 91.4% | 105.7% | 92.9% |
| $f_{10}(x)$ | 99.2% | 92.5% | 92.6% | 92.7% | 97.9% | 92.7% |
| $f_{11}(x)$ | 98.8% | 98.4% | 98.5% | 98.3% | 97.6% | 98.4% |
| $f_{12}(x)$ | 50.1% | 97.2% | 97.4% | 101.2% | 85.4% | 98.3% |
| $f_{13}(x)$ | 128.2% | 84.2% | 84.9% | 97.2% | 104.5% | 103.6% |
| $f_{14}(x)$ | 55.6% | 125.7% | 136.5% | 99.5% | 183.2% | 111.9% |
| $f_{15}(x)$ | 1238.6% | 112.8% | 124.6% | 127.6% | 60.8% | 124.2% |

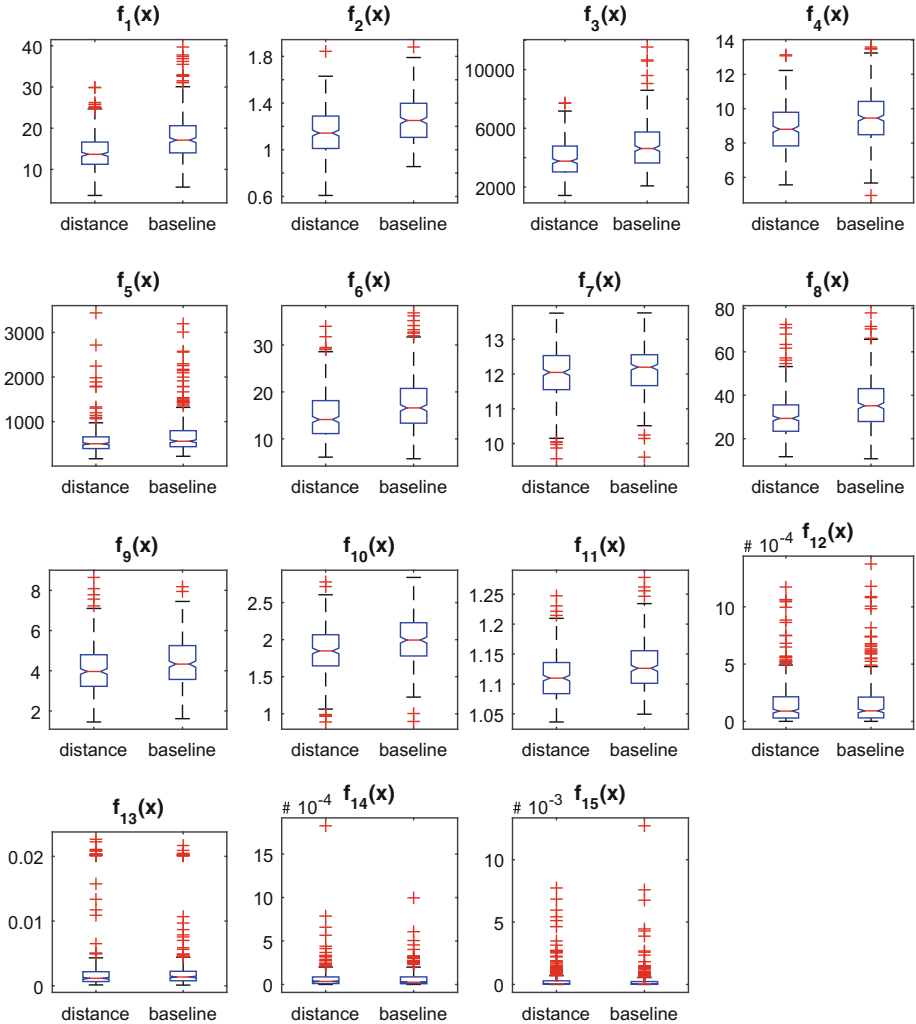


Fig. 3. The boxplots comparing distance-based tournament to the standard tournament selection.

$f_{13}(x)$ to $f_{15}(x)$ is not entirely clear. However, they all have a lower dimensionality in common. The loss of information from mapping low dimension genomes to the fitness might not be as large as in problems with higher dimensionality, dismantling the advantage of the distance-based method.

5 Conclusion

We have presented a new distance-based tournament for genetic algorithms. By influencing the chance of selection into the tournament, chances of being

selected for survival and reproduction are influenced. While this short article is only a proof-of-concept, we think that it shows the potential of influencing the tournament composition through additional information gained from analyzing the population's distribution. The method does improve the performance of the GAs in most cases of a standard benchmark test set. Both the mean and the median of the MSE can be improved by the new method which incorporates the Canberra distances between genomes to construct tournaments in the selection mechanism. In future work we will try to further exploit the information that is in the distribution of the population to improve the search performance of GAs. Further, the technique can also be applied to genetic programming, when the genetic distances are replaced with distances of output vectors. Preliminary experiments show that the technique might be even more promising in this area.

A Appendix

Table 5 shows the quantiles and mean of the mean-squared errors for all the benchmark problems. The baseline results are dominated for nearly all problems except for f_{13} to f_{15} .

Table 5. Mean-squared errors

| Test function | Distance-based | | | | | | Baseline | | | | | |
|---------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | Min | $Q_{.25}$ | $Q_{.5}$ | $Q_{.75}$ | Max | Mean | Min | $Q_{.25}$ | $Q_{.5}$ | $Q_{.75}$ | Max | Mean |
| $f_1(x)$ | 3.67 | 11.24 | 13.65 | 16.64 | 29.90 | 14.38 | 5.69 | 14.01 | 17.09 | 20.61 | 39.71 | 18.04 |
| $f_2(x)$ | 0.61 | 1.01 | 1.14 | 1.29 | 1.84 | 1.16 | 0.86 | 1.11 | 1.25 | 1.40 | 1.88 | 1.26 |
| $f_3(x)$ | 1423.67 | 3028.80 | 3764.47 | 4791.36 | 7726.67 | 3910.02 | 2076.65 | 3635.05 | 4624.08 | 5747.95 | 11542.03 | 4826.56 |
| $f_4(x)$ | 5.56 | 7.83 | 8.80 | 9.79 | 13.14 | 8.86 | 4.95 | 8.49 | 9.45 | 10.42 | 13.59 | 9.47 |
| $f_5(x)$ | 167.17 | 395.45 | 503.46 | 658.47 | 3438.40 | 573.74 | 222.88 | 438.02 | 560.03 | 797.26 | 3195.39 | 703.98 |
| $f_6(x)$ | 6.10 | 11.12 | 14.12 | 18.15 | 33.98 | 14.87 | 5.75 | 13.34 | 16.60 | 20.75 | 36.85 | 17.60 |
| $f_7(x)$ | 9.56 | 11.55 | 12.05 | 12.53 | 13.76 | 12.00 | 9.61 | 11.67 | 12.20 | 12.56 | 13.76 | 12.11 |
| $f_8(x)$ | 11.71 | 23.43 | 29.34 | 35.56 | 72.61 | 30.43 | 10.77 | 27.92 | 35.16 | 43.07 | 77.92 | 36.21 |
| $f_9(x)$ | 1.45 | 3.23 | 3.96 | 4.80 | 8.64 | 4.11 | 1.62 | 3.57 | 4.33 | 5.25 | 8.18 | 4.42 |
| $f_{10}(x)$ | 0.89 | 1.65 | 1.85 | 2.07 | 2.78 | 1.84 | 0.90 | 1.78 | 1.99 | 2.23 | 2.84 | 1.99 |
| $f_{11}(x)$ | 1.04 | 1.08 | 1.11 | 1.14 | 1.25 | 1.11 | 1.05 | 1.10 | 1.13 | 1.16 | 1.28 | 1.13 |
| $f_{12}(x)$ | 2.4E-07 | 3.0E-05 | 9.0E-05 | 2.1E-04 | 1.2E-03 | 1.6E-04 | 4.9E-07 | 3.0E-05 | 9.2E-05 | 2.1E-04 | 1.4E-03 | 1.7E-04 |
| $f_{13}(x)$ | 1.4E-04 | 6.5E-04 | 1.2E-03 | 2.2E-03 | 2.3E-02 | 2.6E-03 | 1.1E-04 | 7.8E-04 | 1.4E-03 | 2.2E-03 | 2.2E-02 | 2.5E-03 |
| $f_{14}(x)$ | 8.4E-08 | 1.1E-05 | 3.5E-05 | 8.7E-05 | 1.8E-03 | 7.5E-05 | 1.5E-07 | 8.6E-06 | 2.6E-05 | 8.7E-05 | 1.0E-03 | 6.7E-05 |
| $f_{15}(x)$ | 4.0E-07 | 1.6E-05 | 5.8E-05 | 2.9E-04 | 7.7E-03 | 4.1E-04 | 3.2E-08 | 1.4E-05 | 4.6E-05 | 2.3E-04 | 1.3E-02 | 3.3E-04 |

References

1. Ackley, D.: A Connectionist Machine for Genetic Hillclimbing, vol. 28. Springer Science & Business Media (2012)
2. Dixon, L.C.W., Szegö, G.P.: The global optimization problem: an introduction. *Towards Global Optim.* **2**, 1–15 (1978)
3. Golberg, D.E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, p. 102. Addison Wesley, Reading (1989)
4. Griewank, A.O.: Generalized descent for global optimization. *J. Optim. Theory Appl.* **34**(1), 11–39 (1981)
5. Homaifar, A., Qi, C.X., Lai, S.H.: Constrained optimization via genetic algorithms. *Simulation* **62**(4), 242–253 (1994)
6. Lance, G.N., Williams, W.T.: Computer programs for hierarchical polythetic classification (similarity analyses). *Comput. J.* **9**(1), 60–64 (1966)
7. Lee, C.-G., Cho, D.-H., Jung, H.-K.: Niching genetic algorithm with restricted competition selection for multimodal function optimization. *IEEE Trans. Magn.* **35**(3), 1722–1725 (1999)
8. Mahfoud, S.W.: Niching methods for genetic algorithms. *Urbana* **51**(95001), 62–94 (1995)
9. Molga, M., Smutnicki, C.: *Test functions for optimization needs* (2005)
10. Mühlenbein, H., Schomisch, M., Born, J.: The parallel genetic algorithm as function optimizer. *Parallel Comput.* **17**(6–7), 619–632 (1991)
11. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. *Comput. J.* **3**(3), 175–184 (1960)
12. Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. *IEEE Trans. Evol. Comput.* **2**(3), 97–106 (1998)
13. Schwefel, H.-P.P.: *Evolution, Optimum Seeking: The Sixth Generation*. Wiley, New York (1993)
14. Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **3**(2), 82–102 (1999)