

NATURE-INSPIRED ALGORITHMS IN REAL-WORLD OPTIMIZATION PROBLEMS

Petr Bujok^a, Josef Tvrdík^a, Radka Poláková^b

University of Ostrava

^aDepartment of Computer Science

^bCentre of Excellence IT4Innovations, Institute for Research and Applications of Fuzzy Modeling,
30. dubna 22, 701 03 Ostrava,
Czech Republic

{petr.bujok, josef.tvrdik, radka.polakova}@osu.cz

Abstract: Eight popular nature inspired algorithms are compared with the blind random search and three advanced adaptive variants of differential evolution (DE) on real-world problems benchmark collected for CEC 2011 algorithms competition. The results show the good performance of the adaptive DE variants and their superiority over the other algorithms in the test problems. Some of the nature-inspired algorithms perform even worse than the blind random search in some problems. This is a strong argument for recommendation for application, where well-verified algorithm successful in competitions should be preferred instead of developing some new algorithms.

Keywords: Global optimization, real-world optimization problems, nature-inspired algorithms, adaptive differential evolution, experimental comparison.

1 Introduction

Solving the global optimization is a frequent part of real-world technological problems. In spite of the progress made in the evolutionary algorithms and development of their adaptive variants in last decades, see e.g. [4, 6], many authors often use either the basic variants of evolutionary algorithms or they try to develop some own variant of nature-inspired algorithm to solve their actual optimization problem.

The main goal of this paper is to demonstrate that such approach is irrational because the advanced evolutionary algorithms available to application are better performing than many basic variants of nature-inspired algorithms. The advanced evolutionary algorithms supported by serious research should be preferred in the applications.

Selected eight nature inspired algorithms are compared with three advanced DE variants and the blind random search on the set of 22 real-world optimization problems collected for algorithm competition held at CEC 2011 [5]. The results show that the nature-inspired algorithms are not competitive with the advanced DE variants and some of them perform even worse than the blind random search.

2 Algorithms Selected to Experimental Comparison

A survey of bio-inspired algorithms has been presented recently in [7]. Authors formed categories and hierarchy of optimization algorithms as: swarm-intelligence-based \subset bio-inspired \subset nature-inspired. This paper and the book [18] were the main sources for the selection of nature-inspired algorithms for this experimental comparison.

The other algorithms were selected using our experience from two experimental studies [1, 2] and the paper [10], where authors studied the performance of the various swarm and evolutionary algorithms in dependence of the time demands using real-world optimization problems CEC 2011 [5].

2.1 Nature-Inspired Algorithms

Artificial bee colony algorithm (*ABC*) was proposed in 2005 [8]. Parameter *limit*, usually equal to the population size, controls a number of unsuccessful new food positions, necessary to find a new random food position. An employed bee position is updated by $y(j) = P(i, j) + (P(i, j) - P(k, j)) U(-1, 1)$, where j is randomly selected index from $(1, D)$ of the position to be updated (D is the dimension of the problem), k is randomly selected bee different from current i th bee and $U(-1, 1)$ is a random number for the uniform distribution with parameters given in parentheses.

Bat algorithm (*Bat*) uses parameter settings that follows the original publication [16]. Maximal and minimal frequencies are set up $f_{\max} = 2$, $f_{\min} = 0$, local-search loudness parameter is initialized $A_i = 0.9$ for each bat-individual and reduced if a new bat position is better than the old one using coefficient $\alpha = 0.95$. Emission

rate parameter is initialized to each bat-individual $r_i = 0.1$ and increased by parameter $\gamma = 0.9$ in the case of successful offspring.

Dispersive flies optimization algorithm (*DFO*) was proposed in 2014 [12], the only control parameter called disturbance threshold, is set to $dt = 1 \times 10^{-3}$.

Cuckoo search algorithm (*Cuckoo*) was introduced in 2009 [19]. Probability of the cuckoos' eggs laid in a bird-host nest is set $pa = 0.25$ and the control parameter of Lévy flight random walk is set to $\lambda = 3/2$.

Firefly algorithm (*Firefly*) proposed in 2008 [18] has several control parameters that are set as follows, randomization parameter $\alpha = 0.5$, light absorption coefficient $\gamma = 1$, and attractiveness is updated using its initial and minimal values $\beta_0 = 1$, $\beta_{\min} = 0.2$.

From the family of the algorithms modelling the life of plants, Flower Pollination Algorithm for Global Optimization (*Flower*) proposed in 2012 [17] was selected to comparison. The main control parameter equals to probability of switching between global and local search is set to $p = 0.8$. A parameter controlling Lévy distribution is set up $\lambda = 3/2$, as in the Cuckoo search algorithm.

Particle swarm optimization (*PSO*) originally proposed in 1995 belongs to very popular and studied nature-inspired algorithms [9]. The basic variant of PSO with slightly enhanced of particles' velocities updating by the variation coefficient w and coefficient c is used in this experimental comparison. The control parameter of variation w is set for each generation as a linear interpolation from maximal value $w_{\max} = 1$ to $w_{\min} = 0.3$. Parameter controlling a local and a global part of the velocity updating is set $c = 1.05$. Velocity is updated by $\mathbf{v}_{i,G+1} = w_{G+1} \mathbf{v}_{i,G} + c U(0, 1) (\mathbf{p}_{\text{best}} - \mathbf{x}_i) + c U(0, 1) (\mathbf{g}_{\text{best}} - \mathbf{x}_i)$, where G denotes generation, $U(0, 1)$ is random number generated from uniform distribution with parameters given in parentheses, \mathbf{x}_i is current particle, \mathbf{p}_{best} is up-to-now best historical position of the current particle, and \mathbf{g}_{best} is the best particle in swarm history.

Self-organizing migration algorithm (*SOMA*) was proposed in 2000 as a model of a pack of predators [20]. Soma has several controls parameters and particles strategies, the best settings based on our preliminary experiments was taken for this experiment. Parameter controlling the length of individual way toward to leader is set $PathLength = 2$, the step size is set to $Step = 0.11$, and perturbation parameter is set $Prt = 0.1$. There are several strategies of individual movement, the best performing strategy *all-to-one* as indicated the preliminary experiments was applied to comparison on CEC 2011 benchmark.

2.2 Other Algorithms Selected for Comparison

Four other algorithms are selected to experimental comparison with nature-inspired algorithms. One of them is the blind random search (*RS*) which was proposed by Rastrigin [11] in 1963. RS is the simplest stochastic algorithm for global optimization. It generates a new trial point from uniform distribution over the search space and compares the function value of the trial point with the best point found so far. If the new trial point is better ($f(\mathbf{x}_{\text{new}}) < f(\mathbf{x}_{\text{best}})$), \mathbf{x}_{new} replaces the old point. No learning mechanism or exploitation of knowledge from previous search is used. The sense of RS inclusion into experiment is to have an algorithm as reference. The algorithms outperformed by RS should not be considered seriously. The goal is to compare nature inspired algorithms with the most naive stochastic optimization approach.

Three algorithms in experimental comparison are adaptive variants of differential evolution proposed recently. The selected algorithms outperform the state-of-the-art DE variants declared in [6] or are competitive with them.

One of them is CoBiDE [15] which uses the covariance-matrix-based crossover and bimodal distribution of control parameters (F and CR) with using the values of the parameters in stochastic manner in order to adapt the parameter setting to the currently solved problem. The exploitation of covariance matrix should increase the efficiency in the optimization problems where coordinates of points in the population are highly correlated.

L-SHADE was proposed by Tanabe and Fukunaga [14] as an improvement of their SHADE [13] algorithm. Population size in L-SHADE is decreased linearly in order to prefer exploration in early stages and exploitation in last stages of the search. Control parameters of F and CR are adapted according to the successes in previous generations. An archive of outperformed old solutions is used for current-to- p best/1 mutation. The results of CEC 2014 show that L-SHADE is highly efficient algorithm capable to outperform many of the state-of-the-art DE variants.

Another enhanced variant of SHADE is also chosen for the experiments. This algorithm called SHADE4 was introduced in [3] for CEC 2016 competition of optimization algorithms. In SHADE4, four different DE strategies compete to generate a new trial point. These strategies are combinations of two kinds of mutation and two types of crossover. A strategy applied to generating the new trial point is selected proportionally to the success in previous generations.

3 Experiments

The test suite of 22 real-world problems selected for CEC 2011 competition in Special Session on Real-Parameter Numerical Optimization [5] is used as a benchmark in the experimental comparison. The functions in the

benchmark differ in the computational complexity and in the dimension of the search space which varies from $D = 1$ to $D = 240$, the dimensionality of most problems exceeds $D = 20$, see Table 1. The labels of the test problems are taken from [5]. The test functions are described in [5] in detail, including the experimental settings required for the competition. This experimental setting is also used in our experimental comparison. For each algorithm and problem, 25 independent runs were carried out. The run of the algorithm stops if the prescribed number of function evaluation $MaxFES = 150000$ is reached. The point in the terminal population with the smallest function value is the solution of the problem found in the run.

The population size $N = 90$ was used in all the algorithms except L-SHADE algorithm where the population size is not constant during the search but decreases linearly from $N_{init} = 18 \times D$ to $N_{min} = 4$. The other control parameters are set up according to recommendation of authors in their original papers.

4 Results

The minimal function values found in 25 runs are depicted as boxplots in Figure 1 to 3 in order to make visual comparison of the algorithm efficiency easier. The test problems in these figures are sorted from the lowest dimension in Figure 1 to higher dimension up to 3.

We can see in Figure 1 that six problems are solved successfully by most algorithms. There are only three algorithms that failed in these problems, namely DFO, Firefly, and Bat performing worse than the blind random search. Only two problems (T01 and T10) are more difficult and the performance of the algorithms differs substantially but the adaptive DE variants outperformed the other algorithms.

Figure 2 shows different performance of the algorithms on these problems. Adaptive DE variants appear among well-performing algorithms in all the problems, sometimes accompanied by SOMA, PSO, ABC, Cuckoo or Flower. DFO, Bat, Firefly, and RS appear among worst-performing algorithms, DFO, Bat, and Firefly sometimes perform worse than the blind random search.

Boxplots in Figure 3 demonstrate the good performance of adaptive DE variants. ABC, Cuckoo, Flower, PSO, and SOMA are successful in some problems. DFO, Firefly, and Bat are permanently among the bad-performing algorithms, their efficiency is comparable with the blind random search.

The results from Figures 1 to 3 are summarized in Table 2 where the results of Friedman tests are presented. The test was carried out on medians of minimal function values at three stages of the search, namely after $FES = 50,000, 100,000,$ and $150,000$. The medians at final stage of the search are presented in Table 1. The null hypothesis on equivalent efficiency of the algorithms was rejected at the all stages of the search with $p < 5 \times 10^{-6}$. The algorithms in this table are ordered from left to right with respect to their mean rank from Friedman test at the finish of the search, i.e. after reaching $MaxFES=150,000$. The results show the superiority of the LSHADE and SHADE4 followed by CoBiDE and SOMA. The mean ranks of Cuckoo, Flower, PSO, and ABC are approximately equal to the average rank. The performance of DFO, RS, Bat, and Firefly algorithms is deep under average. Notice that the mean ranks of the algorithms after earlier stages of the search differ from the final ones. SHADE4, CoBiDE, and SOMA are better than LSHADE at the first stage of the search.

The medians of function values found in the solutions of the problems are shown in Table 1, the minimal values of median for each problem are underlined. The complete final results of the algorithms were analyzed in more detail. Kruskal-Wallis non-parametric one-way ANOVA test is applied to each test problem. It was found that the performance of the algorithms in comparison differs significantly. The null hypothesis on the same performance is rejected in all the problems with achieved significance level $p < 0.000005$. Multiple Kruskal-Wallis comparison was then also applied. The best performing algorithms significantly different from the followers and mutually with no significant differences are listed in the column "Best" ordered from the best. The worst performing algorithms significantly different from their predecessors and mutually with no significant differences are listed in the column "Worst", ordered from the worst performing algorithm.

5 Conclusion

The results of experimental comparison of eight popular nature-inspired algorithms with the blind random search and three recent variants of adaptive DE demonstrate clearly the superiority the adaptive DE variants in the efficiency.

Despite our expectation, CoBiDE algorithm using covariance-matrix crossover is not the most efficient algorithm in this benchmark of real-world optimization problems. It seems that the frequency of problems with highly correlated coordinates of the parameters is not so large as it is supposed. The superiority of LSHADE in most test problems indicates the importance of population size adaptation in the solution of the real-world problems.

We hope that it is a strong argument for the recommendation for researchers and engineers who need to apply an algorithm to the solution of their real-world optimization problems: Please, do not develop your own modification but use a state-of-the-art algorithm. These algorithms are supported by thorough research and

Table 1: CEC 2011 real-world optimization problems with medians of function values found by the algorithms and results of Kruskal-Wallis multiple comparison

Prob	D	ABC	Bat	Cuckoo	DFO	Firefly	Flower	PSO	SOMA	RS	CoBiDE	LSHADE	SHADE4	Best	Worst
T01	6	13.465	25.799	7.419	27.357	28.411	11.879	12.571	11.708	21.962	0	0	0	LSHA,SHA4,CoBi	DFO,Frfly,Bat
T02	30	-18.198	-3.702	-9.426	-6.556	-12.680	-7.808	-15.823	-18.730	-4.007	-16.187	-25.071	-24.118	LSHA,SHA4	Bat,RS,DFO
T03	1	1.15E-05	almost all	Frfly											
T04	1	0	0	0	0	0	0	0	0	0	0	0	0	almost all	Frfly
T05	30	-34.914	-16.492	-31.933	-9.828	-18.228	-30.277	-31.128	-32.381	-18.434	-33.090	-36.559	-36.544	SHA4,LSHA,ABC	DFO,Bat
T06	30	-22.990	-5.955	-22.953	0.000	0.000	-16.818	-23.005	-27.430	-12.884	-26.557	-29.138	-29.166	SHA4,LSHA	DFO,Frfly,Bat
T07	20	1.459	1.971	1.203	2.578	2.371	1.232	1.287	1.119	1.760	1.228	1.046	0.611	SHA4,LSHA	DFO,Frfly
T08	7	220	1015	220	282	351	220	220	220	220	220	220	220	almost all	Bat,Frfly,DFO
T09	126	1.34E+04	2.51E+05	3.59E+05	4.64E+05	3.18E+06	6.24E+05	5.52E+05	2.22E+05	2.57E+06	5.64E+04	2.86E+03	1.98E+03	SHA4,LSHA,ABC	Frfly,RS
T10	12	-16.283	-8.044	-19.773	-8.840	-7.597	-15.666	-15.477	-16.862	-8.617	-21.711	-21.644	-21.842	SHA4,CoBi,LSHA	Bat,Frfly,RS,DFO
T11.1	120	5.79E+04	1.07E+08	5.58E+05	9.73E+07	2.50E+06	1.82E+06	1.53E+06	1.27E+06	2.12E+08	1.28E+05	5.22E+04	5.25E+04	LSHA,SHA4,ABC	RS,Bat,DFO
T11.2	240	1.75E+06	1.01E+07	2.07E+06	5.75E+06	7.33E+06	3.80E+06	5.31E+06	3.64E+06	1.24E+07	1.08E+06	1.08E+06	1.07E+06	SHA4,LSHA,CoBi	RS,Bat
T11.3	6	1.55E+04	1.97E+04	1.54E+04	1.81E+04	5.30E+04	1.54E+04	1.54E+04	1.55E+04	1.57E+04	1.54E+04	1.54E+04	1.54E+04	SHA4,LSHA,	Frfly,Bat,DFO
T11.4	13	1.94E+04	1.93E+04	1.89E+04	1.92E+04	1.92E+04	1.84E+04	1.88E+04	1.92E+04	1.94E+04	1.81E+04	1.81E+04	1.81E+04	CoBi,Flower	ABC,RS,
T11.5	15	3.32E+04	4.75E+05	3.30E+04	2.52E+05	4.96E+05	3.29E+04	3.31E+04	3.29E+04	1.78E+05	3.28E+04	3.27E+04	3.27E+04	LSHA,SHA4,CoBi	Frfly,Bat,DFO
T11.6	40	1.41E+05	2.69E+05	1.37E+05	1.65E+05	7.78E+07	1.32E+05	1.41E+05	1.33E+05	3.25E+06	1.27E+05	1.26E+05	1.30E+05	LSHA,CoBi	Frfly,RS
T11.7	140	3.51E+08	1.33E+10	2.34E+06	1.28E+10	1.48E+10	1.96E+06	2.22E+06	1.98E+06	7.37E+09	1.92E+06	1.92E+06	1.90E+06	SHA4,LSHA,CoBi	Frfly,Bat,DFO
T11.8	96	2.63E+07	8.50E+07	7.83E+06	8.37E+06	1.47E+08	1.41E+06	1.25E+06	9.80E+05	9.00E+07	1.13E+06	9.39E+05	9.40E+05	LSHA,SHA4	Frfly,RS,Bat
T11.9	96	2.53E+07	9.11E+07	9.29E+06	1.25E+07	1.58E+08	1.89E+06	1.45E+06	1.32E+06	9.37E+07	1.56E+06	9.43E+05	9.77E+05	LSHA,SHA4	Frfly,RS,Bat
T11.10	96	2.67E+07	8.50E+07	8.48E+06	1.14E+07	1.48E+08	1.26E+06	1.22E+06	9.88E+05	9.28E+07	1.13E+06	9.39E+05	9.40E+05	LSHA,SHA4	Frfly,RS,Bat
T12	26	19.684	54.984	21.704	75.861	46.122	24.036	19.524	18.042	40.235	18.126	16.320	18.572	LSHA,SOMA	Bat,Frfly,DFO
T13	22	20.964	50.134	25.825	58.816	39.824	25.336	26.693	22.816	39.914	12.618	14.579	14.905	CoBi,LSHA,SHA4	DFO,Bat,RS

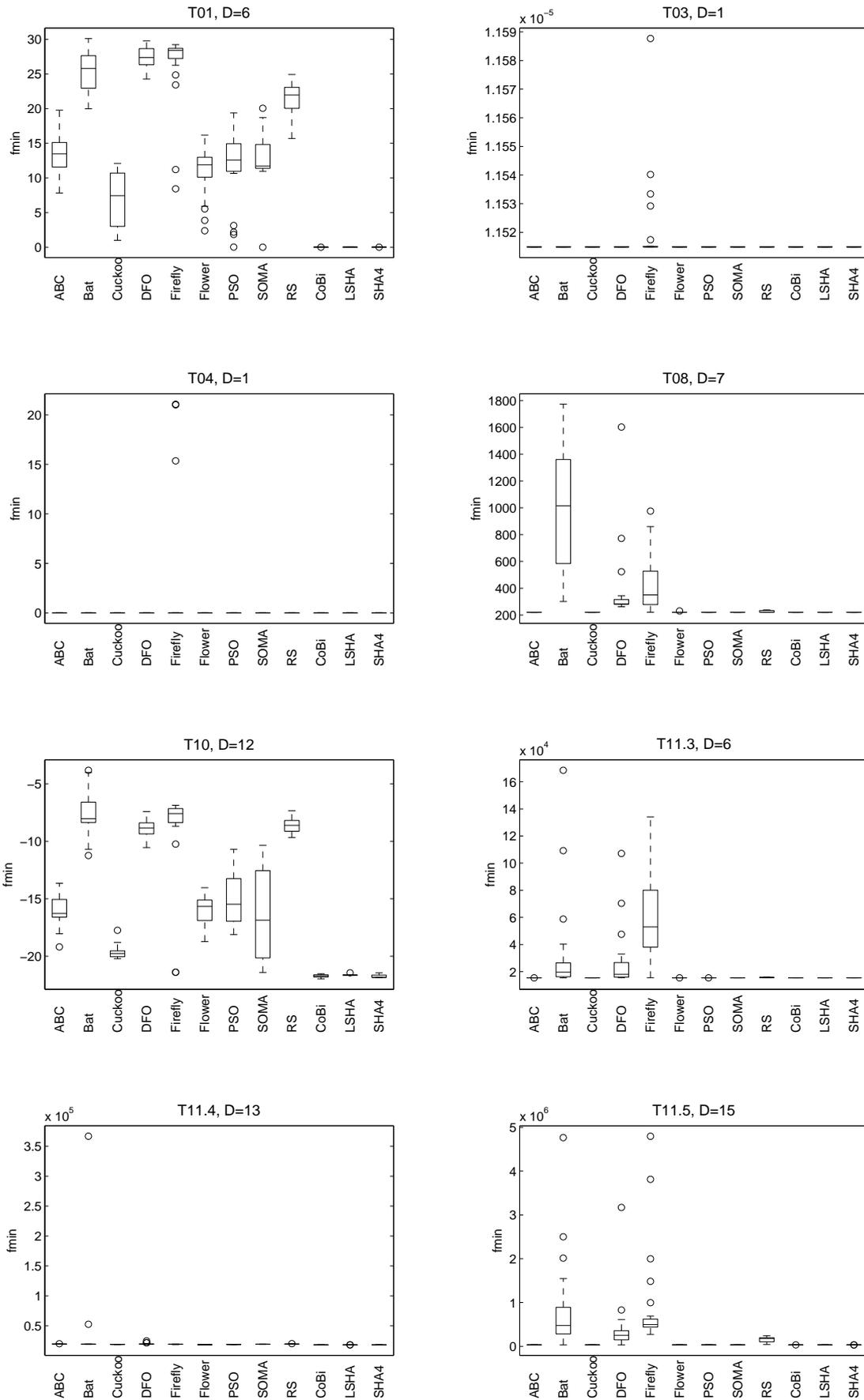


Figure 1: Minimal function values from 25 runs in problems with $D \leq 15$

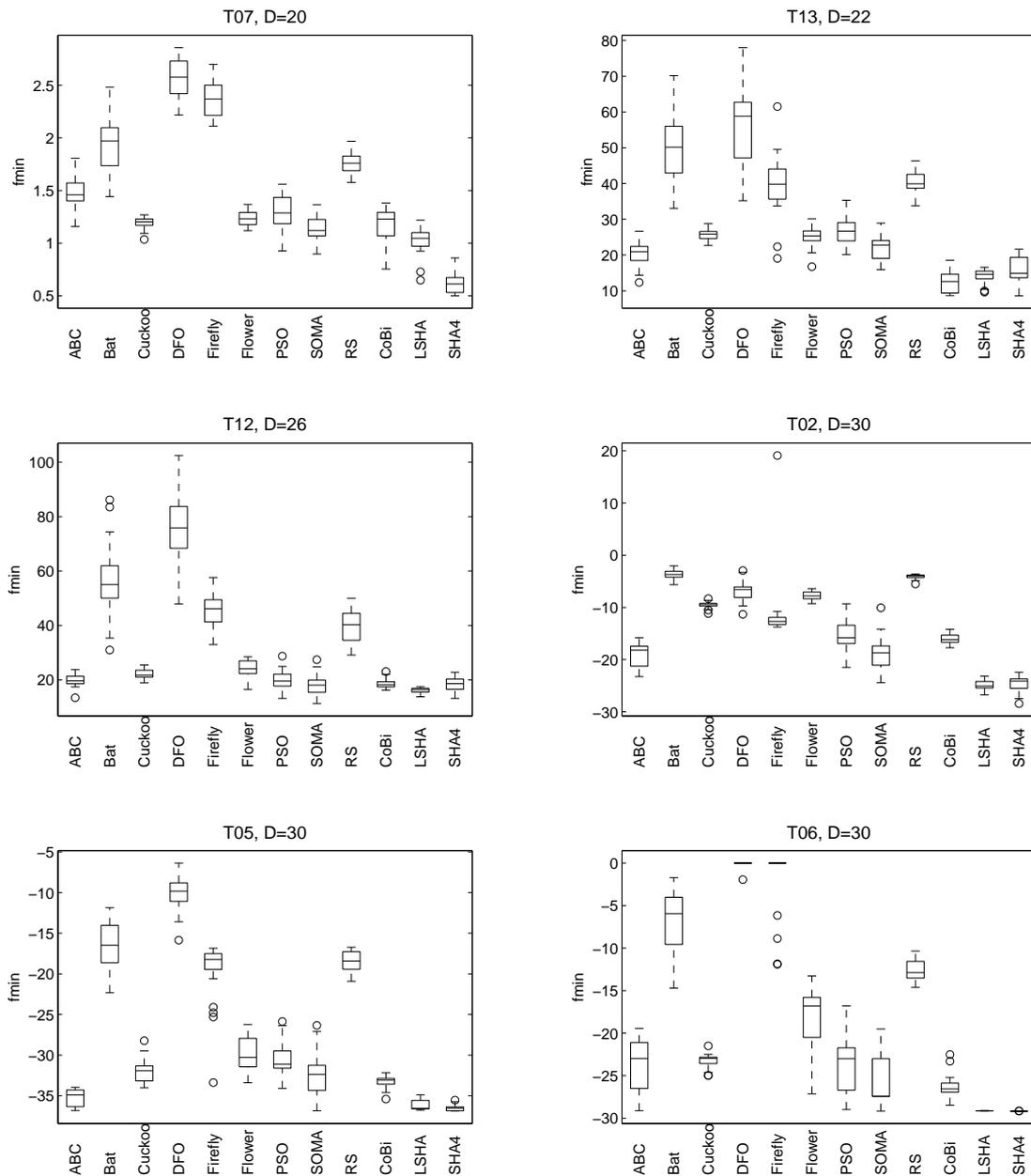


Figure 2: Minimal function values from 25 runs in problems with $20 \leq D \leq 30$

Table 2: Mean ranks from Friedman tests at three stages of the search, $FES = 50000, 100000, 150000$

FES	LSHA	SHA4	CoBiDE	SOMA	Cuckoo	Flower	PSO	ABC	DFO	RS	Bat	Firefly
50,000	5.3	1.9	3.8	4.0	6.3	6.3	5.8	5.6	9.5	10.0	9.6	10.0
100,000	4.4	2.0	3.5	4.4	6.2	6.3	5.5	6.1	9.3	9.9	10.0	10.3
150,000	2.1	2.5	3.7	4.7	6.0	6.4	6.4	6.5	9.5	9.6	10.1	10.5

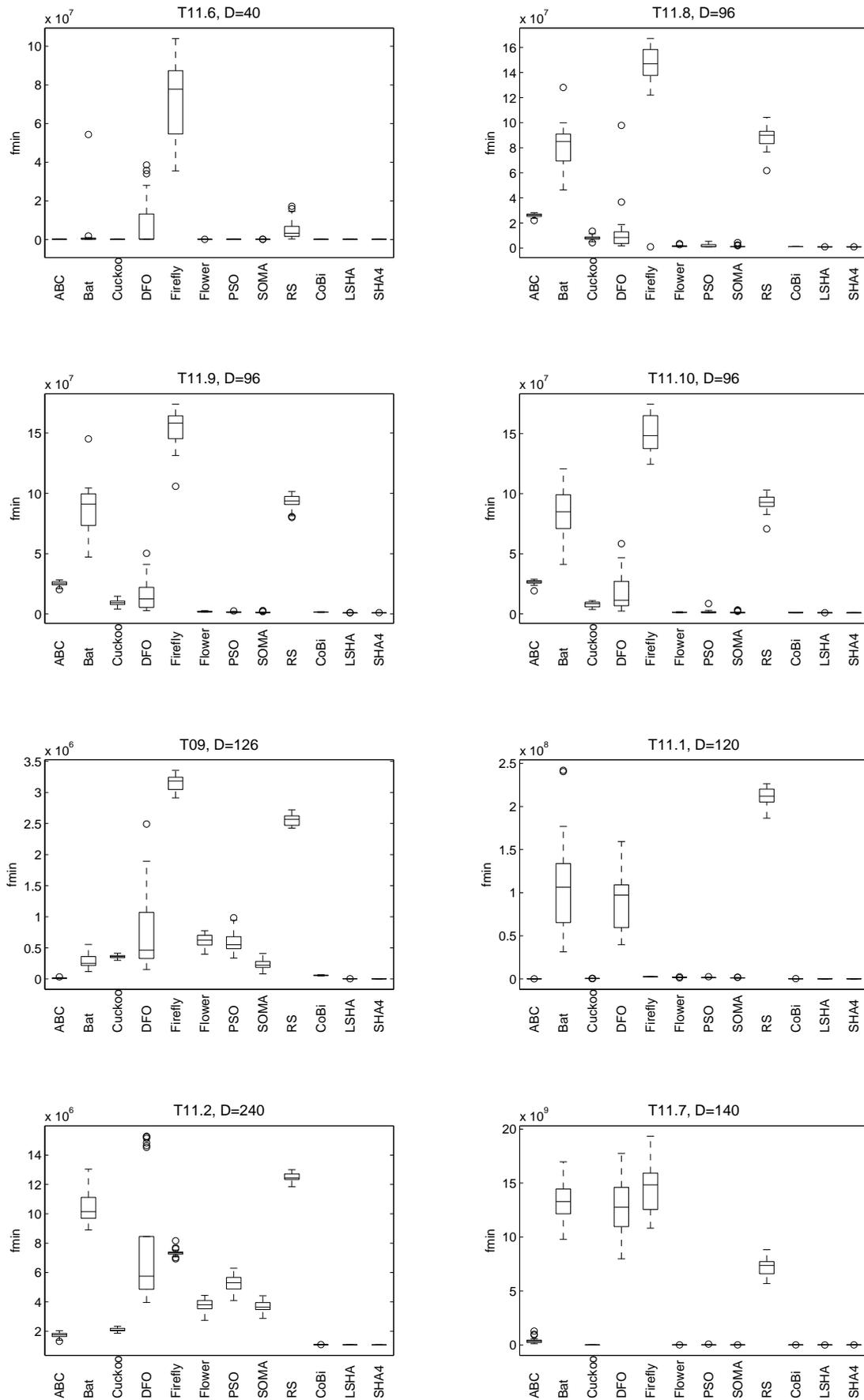


Figure 3: Minimal function values from 25 runs in problems with $D \geq 40$

experimental comparison. You need not implement the selected algorithm themselves frequently. Source codes of many algorithms are available at web sites of their authors or other researchers. For example, the source code in Matlab of three algorithms used in this study is available at www1.osu.cz/~bujok/, the source code of some other state-of-the-art nature-inspired algorithms can be found on web site of MathWorks, www.mathworks.com.

References

- [1] Bujok, P., Tvrdík: Enhanced success-history based parameter adaptation for differential evolution and real-world optimization problems. In: G. Papa, M. Mernik (eds.) *BIOMA, Bioinspired Optimization Methods and their Applications*, Bled, Slovenia, pp. 159–171 (2016)
- [2] Bujok, P., Tvrdík, J., Poláková, R.: Differential evolution with exponential crossover revisited. In: R. Matoušek (ed.) *MENDEL, 22nd International Conference on Soft Computing*, Brno, Czech Republic, pp. 17–24 (2016)
- [3] Bujok, P., Tvrdík, J., Poláková, R.: Evaluating the performance of shade with competing strategies on cec 2014 single-parameter test suite. In: *2016 IEEE Congress on Evolutionary Computation CEC*, pp. 5002–5009 (2016)
- [4] Das, S., Mullick, S., Suganthan, P.: Recent advances in differential evolution-an updated survey. *Swarm and Evolutionary Computation* **27**, 1–30 (2016)
- [5] Das, S., Suganthan, P.N.: Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Tech. rep., Jadavpur University, India and Nanyang Technological University, Singapore (2010)
- [6] Das, S., Suganthan, P.N.: Differential evolution: A survey of the state-of-the-art. *IEEE Transactions on Evolutionary Computation* **15**, 27–54 (2011)
- [7] Fister Jr., I., Yang, X.S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. *Elektrotehnikski vestnik* **80**(3), 116–122 (2013)
- [8] Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Kayseri, Turkey (2005)
- [9] Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *1995 IEEE International Conference on Neural Networks Proceedings, Vols 1-6*, pp. 1942–1948. IEEE, Neural Networks Council (1995). *1995 IEEE International Conference on Neural Networks ICNN 95, UNIV W AUSTRALIA, PERTH, AUSTRALIA, NOV 27-DEC 01, 1995*
- [10] Piotrowski, A.P., Napiorkowski, M.J., Napiorkowski, J.J., Rowinski, P.M.: Swarm intelligence and evolutionary algorithms: Performance versus speed. *Information Sciences* **384**, 34–85 (2017)
- [11] Rastrigin, L.: Convergence of random search method in extremal control of many-parameter system. *Automation and remote control* **24**(11), 1337–1342 (1964)
- [12] al Rifaie, M.M.: Dispersive flies optimisation. In: *Federated Conference on Computer Science and Information Systems, 2014, ACSIS-Annals of Computer Science and Information Systems*, vol. 2, pp. 529–538 (2014). *Federated Conference on Computer Science and Information Systems FedCSIS, Warsaw, POLAND, SEP 07-10, 2014*
- [13] Tanabe, R., Fukunaga, A.: Success-history based parameter adaptation for differential evolution. In: *IEEE Congress on Evolutionary Computation (CEC), 2013*, pp. 71–78 (2013)
- [14] Tanabe, R., Fukunaga, A.S.: Improving the search performance of shade using linear population size reduction. In: *2014 IEEE Congress on Evolutionary Computation CEC 2014*, pp. 1658–1665 (2014)
- [15] Wang, Y., Li, H.X., Huang, T., Li, L.: Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Applied Soft Computing* **18**, 232–247 (2014)
- [16] Yang, X.S.: A new metaheuristic bat-inspired algorithm. In: J. Gonzalez, D. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (eds.) *Nicso 2010: Nature Inspired Cooperative Strategies for Optimization, Studies in Computational Intelligence*, vol. 284, pp. 65–74. Univ Laguna; Carnary Govt; Spanish Govt (2010). *International Workshop on Nature Inspired Cooperative Strategies for Optimization NICSO 2008, Tenerife, SPAIN, 2008*
- [17] Yang, X.S.: Flower pollination algorithm for global optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* **7445 LNCS**, 240–249 (2012)
- [18] Yang, X.S.: *Nature-Inspired Optimization Algorithms*. Elsevier (2014)
- [19] Yang, X.S., Deb, S.: Cuckoo search via Lévy flights. In: *2009 World Congress on Nature Biologically Inspired Computing NaBIC*, pp. 210–214 (2009)
- [20] Zelinka, I., Lampinen, J.: Soma – self organizing migrating algorithm. In: R. Matousek (ed.) *MENDEL, 6th International Conference On Soft Computing*, Brno, Czech Republic, pp. 177–187 (2000)