2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504720

Novel Zigzag-based Benchmark Functions for Bound Constrained Single Objective Optimization

Jakub Kudela
Institute of Automation and Computer Science
Brno University of Technology
Brno, Czech Republic
Email: Jakub.Kudela@vutbr.cz

***Abstract*—The development and comparison of new optimization methods in general, and evolutionary algorithms in particular, rely heavily on benchmarking. In this paper, the construction of novel zigzag-based benchmark functions for bound constrained single objective optimization is presented. The new benchmark functions are non-differentiable, highly multimodal, and have a built-in parameter that controls the complexity of the function. To investigate the properties of the new benchmark functions two of the best algorithms from the CEC'20 Competition on Single Objective Bound Constrained Optimization, as well as one standard evolutionary algorithm, were utilized in a computational study. The results of the study suggest that the new benchmark functions are very well suited for algorithmic comparison.**

I. Introduction

Benchmarking has a crucial role in the development of novel search algorithms as well as in the assessment and comparison of contemporary algorithmic ideas [1]. One of the subclasses of the derivative-free optimization methods are Evolutionary Algorithms (EAs), which proved to be very powerful for solving black-box optimization problems. However, because EAs generally lack theoretical performance results, their development and performance comparison rely mainly on benchmarking. Benchmarking experiments are set up for performance comparison on given problem classes and should support the selection of a suitable algorithm for a given real-world application [2]. Benchmarks are also used to qualify the theoretical predictions of the behaviour of algorithms [3].

There are two main lines of development in benchmarking for EAs, the IEEE Congress on Evolutionary Computation (CEC) competitions [4] and the Comparing Continuous Optimizer (COCO) benchmark suite [6]. The COCO suite is a platform for comparing unconstrained continuous optimizers for numerical optimization. An advantage of the COCO platform is a large number of algorithm results available for comparison. Up to now, 231 distinct (classical as well as contemporary) algorithms have been tested on the COCO suite. On the other hand, the competitions that are organized every year during the CEC aim to compare state-of-the-art stochastic search algorithms. The CEC competitions provide specific test environments for algorithm assessment and comparison. The benchmark functions are constructed from a set of popular benchmark functions, such as the Rastrigin's function, Rosenbrock's function, Griewank's function, Ackley's function, Schwefel's function, and others [4]. A tunable benchmark function for combinatorial problems was recently introduced in [5].

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic project No. CZ.02.1.01/0.0/0.0/16_026/0008392 "Computer Simulations for Effective Low-Emission Energy" and by IGA BUT: FSI-S-20-6538.

In this paper, we propose novel zigzag-based benchmark functions for bound constrained single objective optimization, that are non-differentiable and highly multimodal. The rest of the paper is organized as follows. Section II introduces the individual components of the new benchmark functions and provides insight into their construction. In Section III we report on computational experiments where we compare two state-of-the-art algorithms and one standard EA on a set of problems that utilize the new benchmark functions. The conclusions and future research are described in Section IV.

II. The Novel Benchmark Functions

The new benchmark functions are constructed as follows. First, we devise a "zigzag" function $z(x)$. For given parameters $k > 0, m > 0$ the zigzag function $z(x)$ at a point $x \in \mathbb{R}$ is computed as:

$$z(x) = \begin{cases} m\left(\frac{1}{2} + (-1)^{\lceil kx \rceil}\left(\frac{\lceil kx \rceil + \lfloor kx \rfloor}{2} - kx\right)\right), \text{if } (kx) \notin \mathcal{Z} \\ 0, \quad \text{if } \frac{kx}{2} \in \mathcal{Z} \\ m, \quad \text{otherwise,} \end{cases}$$

where $\frac{2}{k}$ is the period and m is the amplitude of the zigzag function, as depicted in Fig 1. The next step is a construction of a multimodal function $f(x)$, which is a sum of an absolute value of a high degree polynomial with one root in zero and an absolute value function. The scaling of these two parts is such that the function values on the interval $[-200, 200]$ lie between $[0, 200]$ (this allows us to compose the function with itself any number of times without running into severe numerical difficulties). The reason we care about the behaviour of the function on the interval $[-200, 200]$, and not just the interval $[-100, 100]$ where the optimization will be carried out, is because the benchmark functions will include a shift (and a rotation/scaling). The "polynomial" part of the function f is

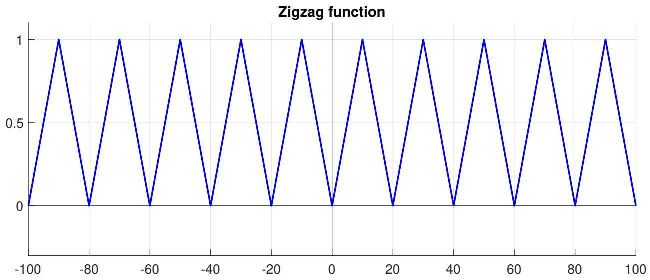


Fig. 1. Zigzag function $z(x)$ with $k = 0.1, m = 1$.

then multiplied with the zigzag function $z(x)$. The particular choice for the function $f(x)$ in this paper is the following:

$$f(x)=3{\cdot}10^{-9}|(x-50)(x-190)x(x+70)(x+180)|z(x) + 0.2|x|$$

where the individual parts of the function are shown in Fig. 2, and the impact of varying the parameter k of the zigzag function $z(x)$ is shown in Fig. 3. Fig. 3 also shows the structure of the function $f(x)$ composed with itself, i.e. the function $f(f(x))$, for different values of k. The function f has a single global optimum point in 0, is non-differentiable and highly multimodal (the "degree of multimodality" depending on the parameter k).

Finally, the two proposed benchmark functions $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$, for $\mathbf{x} = [x_1, \dots, x_D]^T$ and $\mathbf{x} \in [-100, 100]^D$, are the following:

$$f_1(\mathbf{x}) = \sum_{i=1}^{D} f(x_i)$$
$$F_1(\mathbf{x}) = f_1(\mathbf{M_1}(\mathbf{x} - \mathbf{s_1}))$$
$$f_2(\mathbf{x}) = \sum_{i=1}^{D} f(f(x_i))$$
$$F_2(\mathbf{x}) = f_2(\mathbf{M_2}(\mathbf{x} - \mathbf{s_2}))$$

where $\mathbf{s_1}, \mathbf{s_2} \in [-100, 100]^D$ are random shifts of the optimal solution and $\mathbf{M_1}, \mathbf{M_2}$ are random rotation/scaling matrices, with eigenvalues in the range $[0.5, 1]$. The contour and surface plots of $F_1(x)$ and $F_2(x)$ for $D = 2$ and different values of the parameter k can be seen in Fig. 4. The rotation/scaling matrices were constructed in the following way: for a given dimension D, we generate a random square matrix $\mathbf{A}$, and construct a matrix $\mathbf{B} = \mathbf{A}'\mathbf{A}$. Then we get matrices $\mathbf{P}, \mathbf{R}, \mathbf{Q}$ from the singular value decomposition of $\mathbf{B}$, i.e. $\mathbf{B} = \mathbf{PQR}'$. Lastly, we generate a D dimensional vector $\mathbf{v}$ whose individual components are uniformly distributed random values on the interval [0.5,1], and we construct the matrix $\mathbf{M}$ as $\mathbf{M} = \mathbf{P} \cdot \mathrm{diag}(\mathbf{v}) \cdot \mathbf{R}'$, where diag$(\cdot)$ transfors a vector into a diagonal matrix. This ensures that the eigenvalues of $\mathbf{M}$ lie on the interval [0.5,1]. The rotation/scaling matrix is an integral part of the benchmark function [7], as it creates additional difficulty for the optimization algorithms [8].

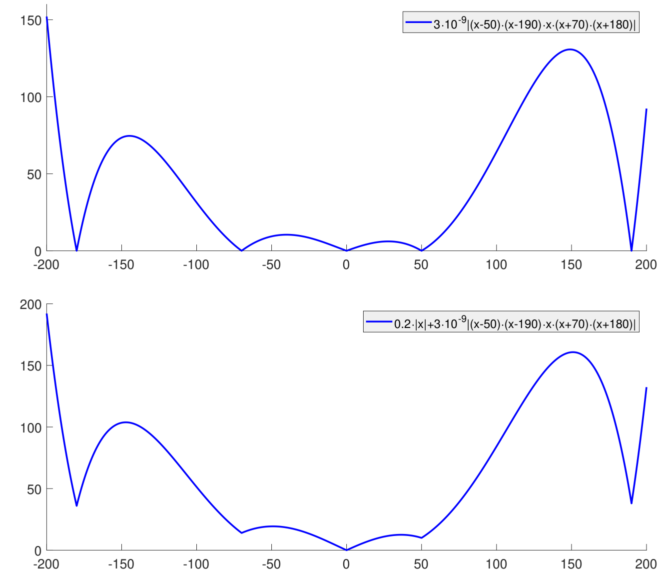


Fig. 2. Partial construction of the function $f(x)$.

III. Computational Experiments

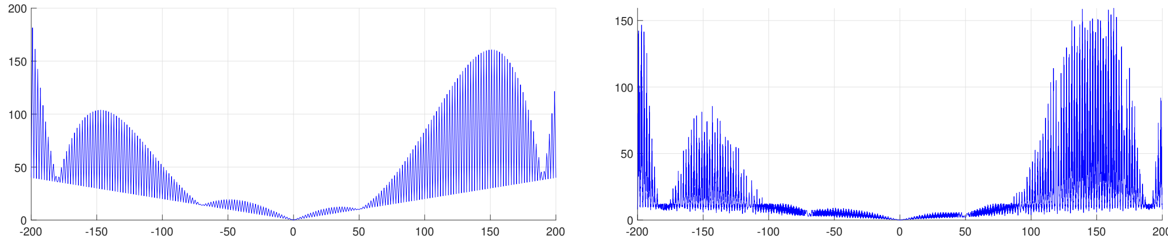
A. Optimization Algorithms and Experimental Settings

The first algorithm we chose for the computational comparison, is the canonical particle swarm optimization (PSO) algorithm that simulates swarm behaviors of social animals such as the bird flocking or fish schooling [9]. The particular implementation and parameter setting for the PSO is the one that was shipped along with the benchmark suite for the CEC'20 Competition on Single Objective Bound Constrained Optimization [4].

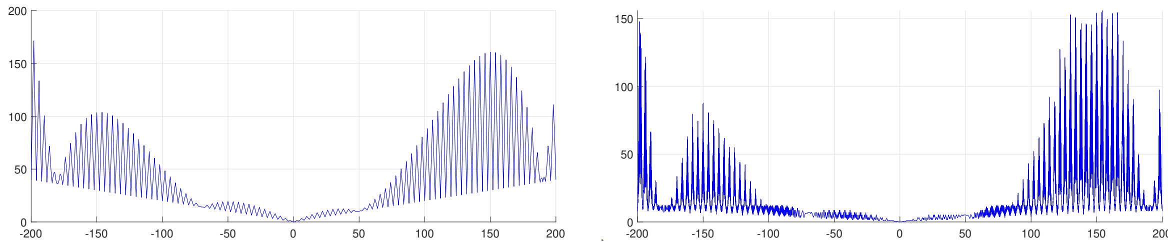
The second algorithm for the comparison is the winner of the CEC'20 Competition on Single Objective Bound Constrained Optimization, the Improved Multi-operator Differential Evolution (IMODE) algorithm [10]. This algorithm utilizes multiple differential evolution operators and a sequential quadratic programming local search procedure for accelerating its convergence.

The third algorithm for the comparison is the runner-up of the same competition, the Adaptive Gaining-Sharing Knowledge (AGSK) based algorithm [11]. This algorithm extends and improves the original GSK [12] algorithm by adding adaptive settings to the two important control parameters: the knowledge factor and the knowledge ratio, which control junior and senior gaining and sharing phases between the solutions during the optimization loop.

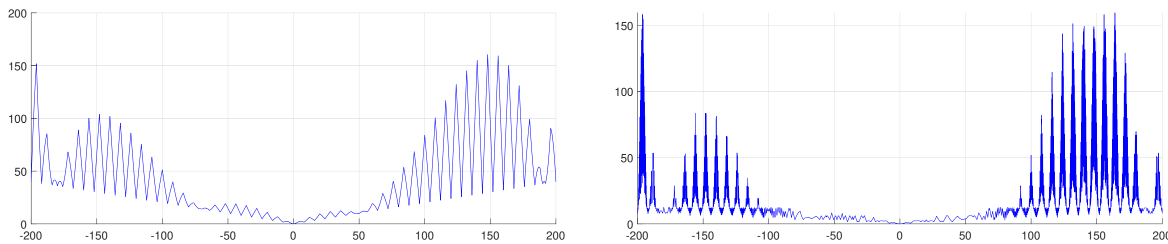
We use the same benchmark rules as the CEC'20 competition: the three algorithms are evaluated on the two benchmark functions with $D = [5, 10, 15, 20]$ dimensions, parameter $k = [2^0, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-4}]$, and a search space of $[-100, 100]^D$. As a change in the parameter m can be though of as a scaling of the polynomial part of the function f, we set it to $m = 1$. The maximum number of function evaluations were set to 50,000, 1,000,000, 3,000,000 and 10,000,000 fitness function evaluations for problems with $D = [5, 10, 15, 20]$, respectively.



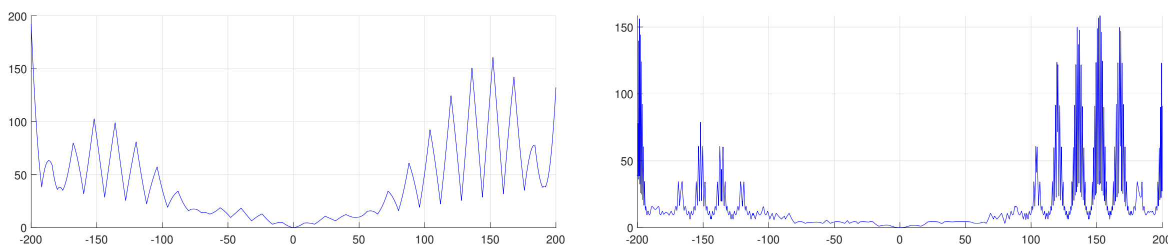
$f(x)$ and $f(f(x))$ for $k = 2^0$



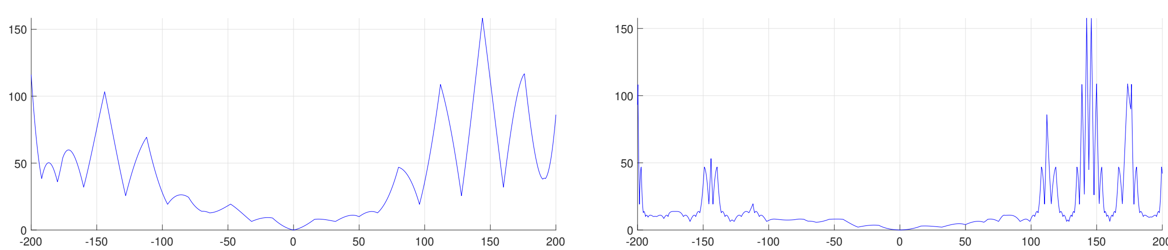
$f(x)$ and $f(f(x))$ for $k = 2^{-1}$



$f(x)$ and $f(f(x))$ for $k = 2^{-2}$



$f(x)$ and $f(f(x))$ for $k = 2^{-3}$



$f(x)$ and $f(f(x))$ for $k = 2^{-4}$

Fig. 3. Functions $f(x)$ (left) and $f(f(x))$ (right) for different values of parameter k

.

All algorithms were run 30 times to obtain representative results. For every run, if the function value of the solution was less than or equal to 1e–8, it was set as zero. The particular values of $\mathbf{M}$ and $\mathbf{s}$ can be found in the authors github[1].

For both IMODE and AGSK, we use the same parameter settings that they used in the CEC'20 competition [13]. The algorithms were run in a MATLAB R2020b, on a PC with 3.2 GHz Core I5 processor, 16 GB RAM, and Windows 10.

[1]https://github.com/JakubKudela89/Zigzag

B. Results

The results of the computational experiments with the three algorithms are summarized in Table I for the benchmark function $F_1(\mathbf{x})$, and Table II for the benchmark function $F_2(\mathbf{x})$. In both tables, we report the best value over the 30 independent runs (min), the median value, the mean value, the worst value (max), and the standard deviation (std). The best result of the three algorithms in the categories min, median, mean, and max is highlighted for each problem instance.

Firstly, it is clear from the results that both the benchmark functions are not "impossible" to optimize, as there were plenty of instances where the algorithms found the optimal solution. However, the instances are not "too easy" so that the algorithms find the optimum reliably – this, in our opinion, makes these benchmark functions worth investigating. Another observation to be made is that for both test functions, reducing the zigzag parameter k really reduces the complexity of the problems. In particular, the changes from $k = 2^{-2}$ to $k = 2^{-3}$ and then to $k = 2^{-4}$ seem to have the biggest impact, while the statistical results for problems with $k = [2^0, 2^{-1}, 2^{-2}]$ are relatively stable (for the same dimension D). Unsurprisingly, the difficulty of the test problems also increases with the dimension D.

The most surprising results come from the comparison of the three algorithms. Let us first focus on the first benchmark function $F_1(\mathbf{x})$. In dimension $D = 5$, the most successful algorithm was the PSO, with both AGSK and IMODE being weaker (without any noticeable difference between them) for all instances but the most simple one with $k = 2^{-4}$. For $D = 10$, the situation is a bit different – while PSO again dominated the difficult instances $k = [2^0, 2^{-1}, 2^{-2}, 2^{-3}]$, it was IMODE that was best for the instance $k = 2^{-4}$, but AGSK performed better that IMODE on the instances $k = [2^0, 2^{-1}, 2^{-2}, 2^{-3}]$. For $D = 15$, AGSK is the worst of the three on all instances, with PSO dominating for $k = [2^0, 2^{-1}]$, and IMODE dominating the rest. The results for the largest instances with $D = 20$ have PSO being the best algorithm for all but the simplest instance $k = 2^{-4}$, where it is on the same level as IMODE. AGSK and IMODE behave similarly for $k = [2^0, 2^{-1}, 2^{-2}]$ while IMODE is clearly better for $k = [2^{-3}, 2^{-4}]$.

The results for the second benchmark function $F_2(\mathbf{x})$ are somewhat similar. PSO again dominates the difficult instances for $k = [2^0, 2^{-1}, 2^{-2}, 2^{-3}]$ for all dimension but for $D = 15$, where IMODE seems to work a bit better. On the simplest instance $k = 2^{-4}$, IMODE is best in dimensions $D = [15, 20]$, with $D = 10$ having AGSK as the winner, and $D = 5$ being solved perfectly by all three algorithms. AGSK is the weakest algorithm of the three in all dimensions, apart from $D = 10$.

Overall, on both of the newly proposed benchmark functions, neither of the two best algorithms from the CEC'20 competition performed significantly better than a standard PSO. We would argue that this a prime reason for investigating these benchmark functions even further and for including them in future competitions and benchmark suits.

IV. Conclusion

In this paper, we presented two novel zigzag-based benchmark functions for bound constrained single objective optimization, which have a simple build-in parameter that can be used to increase their complexity. The construction of these functions is straightforward enough to allow for a wide range of variations, extension, and further study. We also used the two best algorithms from the CEC'20 competition and a standard PSO for computational experiments on test instances utilizing the newly proposed benchmark functions. The results of the experiments suggest that the new benchmark functions are well suited for algorithmic comparison. Future research will encompass comparing a wider selection of algorithms, and developing multimodal benchmark functions [14] using the presented technique.

References

[1] M. H. Hans and G. Beyer, *Benchmarking evolutionary algorithms for single objective real-valued constrained optimization – A critical review*, Swarm and Evolutionary Computation, vol. 44, pp. 927–944, 2019.

[2] O. Mersmann, M. Preuss, H. Trautmann, B. Bischl, and C. Weihs, *Analyzing the BBOB results by means of benchmarking concepts*, Evolutionary Computation, vol. 23, no. 1, pp. 161–185, 2015.

[3] R. L. Rardin and R. Uzsoy, *Experimental evaluation of heuristic optimization algorithms: A tutorial*, Journal of Heuristics, vol. 7, no. 3, pp. 261–304, 2001.

[4] C. T. Yue, K. V. Price, P. N. Suganthan, J. J. Liang, M. Z. Ali, B. Y. Qu, N. H. Awad, and P. P. Biswas, *Problem Definitions and Evaluation Criteria for the CEC 2020 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization*, Tech. Rep., Zhengzhou University and Nanyang Technological University, 2019.

[5] T. Weise and Z. Wu, *Difficult Features of Combinatorial Optimization Problems and the Tunable W-Model Benchmark Problem for Simulating them*, in GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1769–1776, 2018.

[6] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockho, *COCO: a platform for comparing continuous optimizers in a black-box setting*, Optimization Methods and Software, vol. 36, pp. 114–144, 2021.

[7] K. R. Opara, A. A. Hadi, and A. W. Mohamed, *Parametrized Benchmarking: an outline of the idea and a feasibility study*, in GECCO '20: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion, pp. 197–198, 2020.

[8] P. Bujok and R. Polakova, *Eigenvector Crossover in the Efficient jSO Algorithm,* MENDEL, vol. 25, no. 1, pp. 65–72, 2019.

[9] J. Kennedy and R. Eberhart, *Particle swarm optimization*, in Proceedings of ICNN'95 - International Conference on Neural Networks, pp. 1942–1948, 1995.

[10] K. M. Sallam, S. M. Elsayed, R. K. Chakrabortty, and M. J. Ryan, *Improved Multi-operator Differential Evolution Algorithm for Solving Unconstrained Problems*, in 2020 IEEE Congress on Evolutionary Computation (CEC), article no. 19931315, 2020.

[11] A. W. Mohamed, A. A. Hadi, A. K. Mohamed, and N. H. Awad, *Evaluating the Performance of Adaptive GainingSharing Knowledge Based Algorithm on CEC 2020 Benchmark Problems*, in 2020 IEEE Congress on Evolutionary Computation (CEC), article no. 19931514, 2020.

[12] A. W. Mohamed, A. A. Hadi, A. K. Mohamed, *Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm*, International Journal of Machine Learning and Cybernetics, vol. 11, pp. 1501–1529, 2020.

[13] A. Kazikova, M. Pluhacek, and R. Senkerik, *Why Tuning the Control Parameters of Metaheuristic Algorithms Is So Important for Fair Comparison?,* MENDEL, vol. 26, no. 2, pp. 9–16, 2020.

[14] B. Y. Qu, J. J. Liang, Z. Y. Wang, Q. Chen, and P. N. Suganthan, *Novel benchmark functions for continuous multimodal optimization with comparative results*, Swarm and Evolutionary Computation, vol. 26, pp. 23–34, 2016.

TABLE I

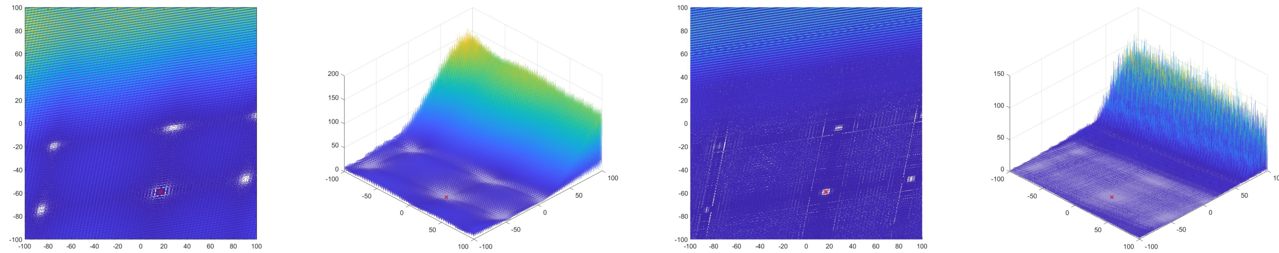
STATISTICS OF THE BEST OBJECTIVE FUNCTION VALUES AFTER THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS OF THE DIFFERENT ALGORITHMS ON THE FIRST BENCHMARK FUNCTION $F_1(\mathbf{x})$ FOR DIFFERENT VALUES OF D AND k. THE ALGORITHM WITH BEST VALUE OVER THE 30 INDEPENDENT RUNS (MIN), THE BEST MEDIAN VALUE, THE BEST MEAN VALUE, AND THE BEST WORST VALUE (MAX) FOR THE PARTICULAR INSTANCE AND IS EMPHASIZED IN BOLD.

		D = 5			D = 10			D = 15			D = 20		
		PSO	AGSK	IMODE	PSO	AGSK	IMODE	PSO	AGSK	IMODE	PSO	AGSK	IMODE
$k = 2^0$	min	**0**	**0**	9.37e–06	1.73e–04	**0**	1.12e+00	**2.80e–01**	2.53e+00	2.70e+00	**1.01e+00**	5.23e+00	6.06e+00
	median	**0**	**0**	1.76e–02	**1.64e–01**	5.22e–01	1.91e+00	**1.91e+00**	5.06e+00	3.90e+00	**2.87e+00**	7.67e+00	8.44e+00
	mean	**0**	1.87e–02	6.98e–02	**4.08e–01**	7.61e–01	1.93e+00	**1.92e+00**	4.89e+00	3.86e+00	**3.18e+00**	7.67e+00	8.51e+00
	max	**0**	2.77e–01	4.08e–01	2.42e+00	**2.37e+00**	2.73e+00	**4.33e+00**	6.31e+00	5.24e+00	**6.94e+00**	9.72e+00	1.03e+01
	std	0	5.47e–02	7.46e–01	6.57e–01	7.48e–01	4.71e–01	1.12e+00	9.45e–01	5.92e–01	1.60e+00	1.15e+00	9.07e–01
$k = 2^{-1}$	min	**0**	**0**	6.75e–08	9.22e–06	**0**	3.68e–01	8.81e–01	**0**	2.64e+00	**1.12e+00**	5.52e+00	7.27e+00
	median	**0**	**0**	8.84e–04	**4.45e–01**	9.04e–01	2.34e+00	**3.38e+00**	4.90e+00	4.02e+00	**3.39e+00**	8.76e+00	8.85e+00
	mean	**0**	6.89e–02	4.23e–02	**6.80e–01**	1.02e+00	2.25e+00	**3.49e+00**	4.80e+00	3.99e+00	**3.60e+00**	8.67e+00	8.88e+00
	max	**2.57e–08**	1.00e+00	2.67e–01	**1.91e+00**	2.74e+00	3.19e+00	6.58e+00	6.11e+00	**5.10e+00**	**7.61e+00**	1.03e+01	1.05e+01
	std	0	2.00e–01	7.98e–02	6.34e–01	7.84e–01	6.64e–01	1.64e+00	1.14e+00	5.72e–01	1.38e+00	1.13e+00	7.54e–01
$k = 2^{-2}$	min	**0**	**0**	**0**	2.14e–03	**0**	1.31e+00	9.60e–01	3.01e+00	**5.40e–01**	**8.02e–01**	6.65e+00	4.25e+00
	median	**0**	1.10e–07	1.34e–04	**6.44e–01**	1.67e+00	2.13e+00	3.47e+00	5.01e+00	**3.16e+00**	**3.37e+00**	8.49e+00	8.08e+00
	mean	3.73e–02	3.25e–02	**2.99e–02**	**6.24e–01**	1.59e+00	2.12e+00	3.70e+00	4.82e+00	**3.04e+00**	**3.53e+00**	8.48e+00	7.91e+00
	max	8.02e–01	**2.87e–01**	5.22e–01	**1.61e+00**	2.99e+00	3.65e+00	7.54e+00	5.72e+00	**4.06e+00**	**7.86e+00**	1.01e+01	9.61e+00
	std	1.55e–01	7.68e–02	1.13e–01	3.26e–01	7.84e–01	5.39e–01	1.86e+00	7.13e–01	7.79e–01	1.30e+00	9.56e–01	1.09e+00
$k = 2^{-3}$	min	**0**	**0**	**0**	1.48e–05	**0**	8.41e–04	**0**	2.94e+00	8.67e–04	**0**	5.96e+00	2.74e+00
	median	**0**	**0**	**0**	**3.79e–02**	3.20e–01	1.55e+00	3.85e+00	4.65e+00	**9.60e–01**	**3.70e+00**	7.91e+00	5.83e+00
	mean	**0**	2.79e–04	1.52e–06	**4.21e–01**	8.00e–01	1.41e+00	3.75e+00	4.58e+00	**8.23e–01**	**3.64e+00**	8.09e+00	5.78e+00
	max	**0**	8.14e–03	2.22e–05	**1.92e+00**	2.74e+00	2.36e+00	8.06e+00	5.71e+00	**1.65e+00**	**7.17e+00**	1.05e+01	7.53e+00
	std	0	1.48e–03	5.48e–06	5.51e–01	9.79e–01	6.59e–01	1.92e+00	6.86e–01	5.20e–01	1.71e+00	1.04e+00	1.28e+00
$k = 2^{-4}$	min	**0**	**0**	**0**	3.97e–06	**0**	**0**	**0**	**0**	**0**	**0**	3.83e+00	1.28e+00
	median	**0**	**0**	**0**	4.59e–03	**0**	**0**	3.20e+00	3.83e+00	**0**	**3.07e+00**	7.60e+00	3.23e+00
	mean	**0**	**0**	**0**	2.00e–01	8.53e–02	**3.49e–07**	3.41e+00	3.62e+00	**0**	3.21e+00	7.37e+00	**2.96e+00**
	max	**0**	**0**	**0**	1.28e+00	1.28e+00	**7.75e–06**	8.88e+00	4.99e+00	**0**	6.12e+00	9.06e+00	**3.97e+00**
	std	0	0	0	4.11e–01	3.24e–01	1.48e–06	2.11e+00	1.19e+00	0	1.60e+00	1.17e+00	7.34e–01

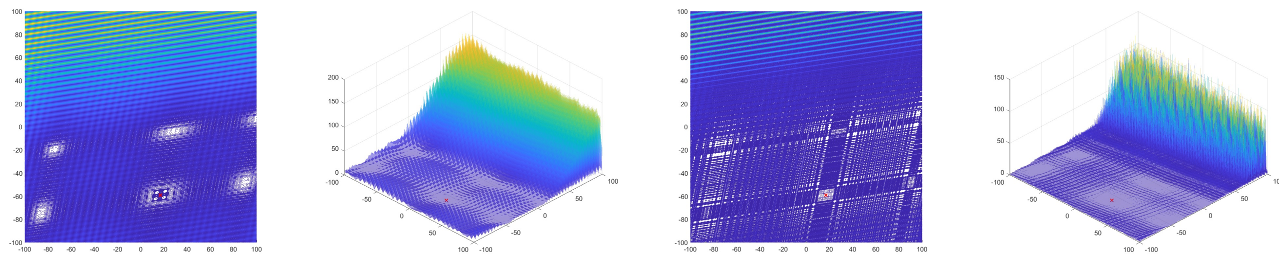
TABLE II

STATISTICS OF THE BEST OBJECTIVE FUNCTION VALUES AFTER THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS OF THE DIFFERENT ALGORITHMS ON THE FIRST BENCHMARK FUNCTION $F_2(\mathbf{x})$ FOR DIFFERENT VALUES OF D AND k. THE ALGORITHM WITH BEST VALUE OVER THE 30 INDEPENDENT RUNS (MIN), THE BEST MEDIAN VALUE, THE BEST MEAN VALUE, AND THE BEST WORST VALUE (MAX) FOR THE PARTICULAR INSTANCE AND IS EMPHASIZED IN BOLD.

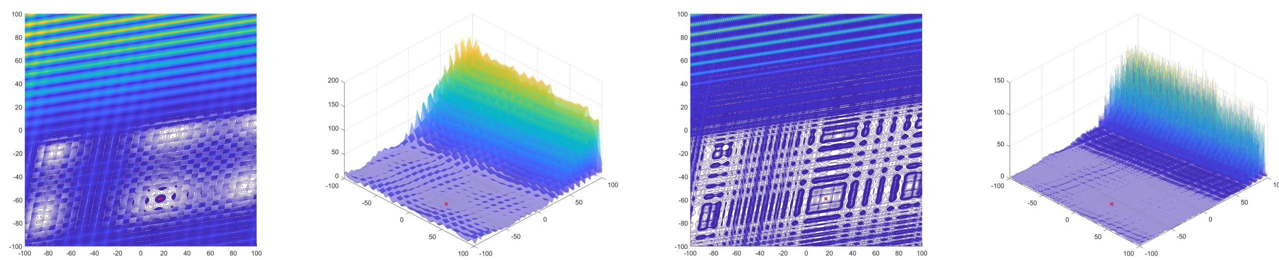
		D = 5			D = 10			D = 15			D = 20		
		PSO	AGSK	IMODE	PSO	AGSK	IMODE	PSO	AGSK	IMODE	PSO	AGSK	IMODE
$k = 2^0$	min	**0**	**0**	4.44e–05	1.70e–05	**0**	1.01e–01	**9.66e–02**	3.64e–01	2.40e–01	**8.69e–02**	6.03e–01	5.26e–01
	median	**0**	6.20e–03	1.20e–02	**1.68e–02**	1.69e–01	2.13e–01	**3.65e–01**	6.85e–01	4.15e–01	**4.75e–01**	9.09e–01	8.01e–01
	mean	**1.75e–02**	2.24e–02	1.84e–02	**4.03e–02**	1.61e–01	2.03e–01	4.91e–01	6.47e–01	**4.00e–01**	**4.99e–01**	8.89e–01	7.81e–01
	max	1.30e–01	1.25e–01	**5.18e–02**	**1.92e–01**	2.87e–01	2.61e–01	1.56e+00	8.40e–01	**4.94e–01**	2.23e+00	1.08e+00	**9.30e–01**
	std	2.95e–02	3.19e–02	1.72e–02	4.95e–02	7.54e–02	4.38e–02	4.34e–01	1.12e–01	6.23e–02	3.87e–01	1.24e–01	9.74e–02
$k = 2^{-1}$	min	**0**	**0**	7.68e–06	3.68e–07	**0**	1.31e–01	**1.11e–02**	3.73e–01	2.90e–01	**5.30e–02**	4.25e–01	6.77e–01
	median	**0**	9.77e–08	1.91e–03	**7.61e–03**	1.22e–01	2.39e–01	**3.16e–01**	6.73e–01	5.40e–01	**3.42e–01**	9.89e–01	1.18e+00
	mean	**0**	3.74e–03	5.26e–03	**2.32e–02**	1.29e–01	2.52e–01	**3.41e–01**	6.44e–01	5.32e–01	**3.81e–01**	9.98e–01	1.16e+00
	max	**0**	3.67e–02	2.81e–02	**1.23e–01**	3.67e–01	4.12e–01	8.91e–01	8.59e–01	**6.68e–01**	**9.25e–01**	1.32e+00	1.37e+00
	std	0	9.20e–03	6.91e–03	3.68e–02	1.08e–01	7.34e–02	2.24e–01	1.21e–01	9.66e–02	2.33e–01	2.14e–01	1.39e–01
$k = 2^{-2}$	min	**0**	**0**	1.32e–08	8.14e–06	**0**	4.56e–02	**1.06e–01**	1.38e–01	1.57e–01	**2.05e–01**	2.69e–01	5.38e–01
	median	**0**	**0**	1.17e–05	**6.24e–03**	1.29e–01	2.12e–01	3.68e–01	4.74e–01	**3.11e–01**	**4.06e–01**	9.12e–01	8.98e–01
	mean	**1.83e–04**	2.44e–03	1.99e–03	**2.82e–02**	1.27e–01	1.96e–01	4.00e–01	4.57e–01	**3.20e–01**	**4.27e–01**	8.84e–01	8.82e–01
	max	**5.49e–03**	2.35e–02	2.71e–02	**2.00e–01**	3.62e–01	3.38e–01	9.49e–01	6.54e–01	**4.37e–01**	**7.04e–01**	1.25e+00	1.10e+00
	std	1.00e–03	5.73e–03	5.61e–03	4.80e–02	8.63e–02	7.96e–02	2.18e–01	1.23e–01	6.62e–02	1.44e–01	1.91e–01	1.33e–01
$k = 2^{-3}$	min	**0**	**0**	**0**	2.67e–08	**0**	4.74e–02	**0**	1.12e–01	1.45e–02	**7.57e–02**	2.93e–01	2.98e–01
	median	**0**	**0**	3.32e–08	**1.38e–04**	7.16e–02	8.30e–02	3.02e–01	2.60e–01	**1.04e–01**	**2.53e–01**	5.87e–01	3.93e–01
	mean	1.08e–03	1.04e–03	**4.02e–06**	**2.84e–02**	7.51e–02	9.08e–02	2.76e–01	2.54e–01	**9.97e–02**	**2.77e–01**	5.52e–01	3.96e–01
	max	3.26e–02	1.42e–02	**4.90e–05**	**1.27e–01**	1.59e–01	1.72e–01	5.15e–01	3.32e–01	**1.63e–01**	**4.94e–01**	7.02e–01	5.05e–01
	std	5.95e–03	3.35e–03	1.07e–05	4.18e–02	5.08e–02	3.00e–02	1.30e–01	5.24e–02	3.69e–02	1.21e–01	9.42e–02	4.94e–02
$k = 2^{-4}$	min	**0**	**0**	**0**	6.19e–08	**0**	**0**	**0**	**0**	**0**	2.19e–02	1.12e–01	**9.46e–05**
	median	**0**	**0**	**0**	4.38e–05	**0**	**0**	8.40e–02	1.53e–01	**0**	1.97e–01	3.05e–01	**1.54e–01**
	mean	**0**	**0**	**0**	2.97e–03	**0**	1.00e–06	1.22e–01	1.47e–01	**0**	2.01e–01	2.91e–01	**1.52e–01**
	max	**0**	**0**	**0**	2.19e–02	**0**	1.12e–05	3.91e–01	2.28e–01	**0**	3.70e–01	3.85e–01	**2.16e–01**
	std	0	0	0	7.57e–03	0	2.68e–06	1.16e–01	4.86e–02	0	7.70e–02	6.51e–02	4.33e–02



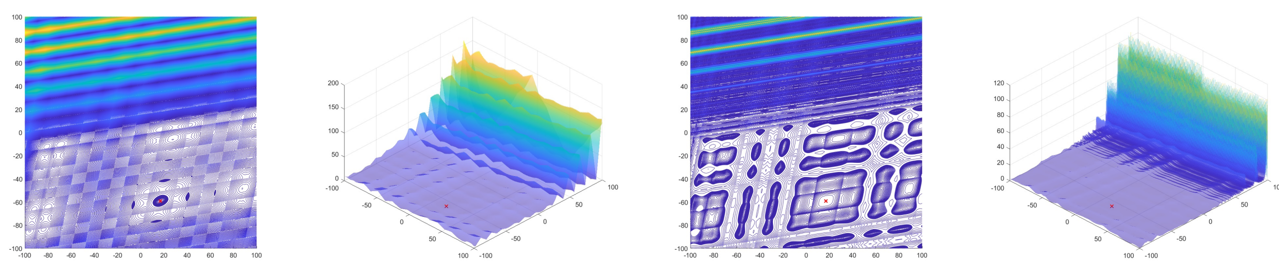
$F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for $k = 2^0$



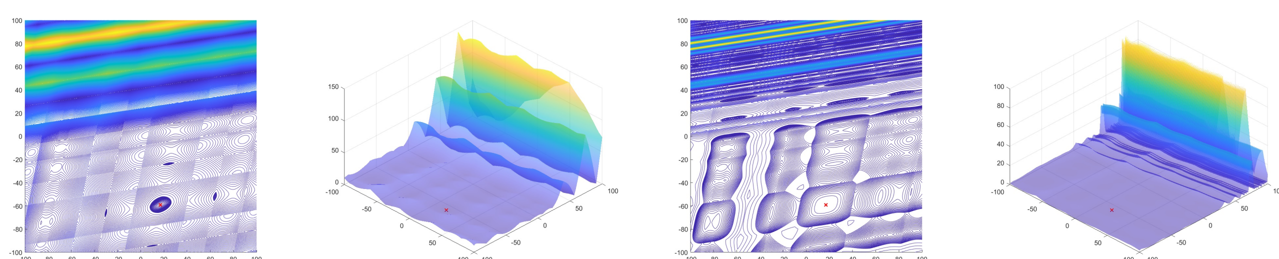
$F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for $k = 2^{-1}$



$F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for $k = 2^{-2}$



$F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for $k = 2^{-3}$



$F_1(\mathbf{x})$ and $F_2(\mathbf{x})$ for $k = 2^{-4}$

Fig. 4. Contour and surface plots of the benchmark functions $F_1(\mathbf{x})$ (left) and $F_2(\mathbf{x})$ (right) for different values of parameter k. The optimum is highlighted by a red marker.