

# Gradient Modification for Cuckoo Search Algorithm to Increase Accuracy

Karolina Kęsik  
Institute of Mathematics  
Silesian University of Technology  
Kaszubska 23, 44-100 Gliwice, Poland  
Email: karola.ksk@gmail.com

**Abstract**—Optimization problems are used in different areas of our life. Very often the functions that are optimized are so complex that classic methods can not deal with them. The solution for it are heuristic techniques that do not guarantee the correct solution in a finite time. In this work, the classic version of Cuckoo Search Algorithm and proposed modifications that allow to increase the precision of obtained solutions are described. The proposal was presented and tested on nine different test functions, and the obtained solutions were discussed in terms of advantages and disadvantages.

## I. INTRODUCTION

More and more often, contemporary problems are described by mathematical equations, which makes it possible to minimize variables in such a way that the cost of production is as low as possible and the highest quality. The large space of solutions, the number of variables or even the landscape of functions can be a big challenge for computers. For this type of tasks, besides the classic methods of finding the best values of the functions are heuristic algorithms. These techniques are inspired by a certain action of physics or nature.

Mathematical models describing the action of nature are created all the time, which is evident in numerous scientific works. In [14], the grasshopper behavior has been described as an optimization algorithm. Again in [10], the authors modeled the way of polar bears survive in arctic condition through the way they move on ice floes and hunt for seals. are tested not only by the optimization problem, but also by the graph. In [2], the authors presented binary monarch butterfly algorithm and it was used to solve knapsack problem. Again in [4], [8], nature inspired algorithm were used as a solver in vehicle routing problem.

In the case of practical applications, such techniques have been used primarily in decision-making systems [19]. In [9], [11], it was used in feature extraction process from images and sound file for the purpose of voice/image recognition. In [3], the heuristics allowed for the construction of commercial microgrids depending on the given assumptions. In a similar way it was used in the assessment of creditworthiness [6] or economic security [5].

Heuristic approach has also found use in location-based social networks [7] or finding outlying points in diabetes data sets [1]. Another important issue is medicine and biomedicine.

Important features describing different lungs diseases on the X-RAY images were detected by heuristic [17], and it were used as support technique for circle detection which allow to detect phases of bacterial life [16].

## II. PROBLEM FORMULATION

The optimization problem is understood as searching for a point whose function value reaches the global extreme, more precisely the minimum or maximum. The choice of maximization or minimization is insignificant because in the case of a function having a maximum, it is enough to negate the function. The same applies to the opposite case. To define the problem formally, let us introduce some signs. For a continuous function  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ , the point  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  will be a global extreme if the following condition is satisfied

$$\min_{\mathbf{x}} f(\mathbf{x}) \quad \text{or} \quad \max_{\mathbf{x}} f(\mathbf{x}). \quad (1)$$

Optimization functions are usually  $n$ -dimensional, for which the extreme is hard to locate. Moreover, classic and numerical methods mostly will not be able to do this. They are called artificial landscape because in three dimensions they resemble familiar shapes from nature or physics. A set of such a functions is presented in Table I, where are defined the most known and representing the basic shapes like bowl-shaped, plate-shaped, valley-shaped, steep ridges and others.

To find the best solution, it is necessary to use heuristic algorithm. It is a technique inspired by the nature, which does not guarantee an accurate solution in a finite time (only approximate).

## III. CUCKOO SEARCH ALGORITHM

In this section, the classic model has been described along with the proposed modifications.

### A. Classic version

Cuckoo Search Algorithm (CSA) was presented in [18] where the author described heuristic technique inspired by the cuckoos behavior while tossing their eggs to the nests of other birds. For obvious reasons, the mathematical model had to be simplified in order to reduce the number of unknown parameters as well as to minimize the number of calculations. To this end, several simplifications have been introduced

Table I: Selected test function for described optimization problem.

Name	Function	Input domain	$\mathbf{x}$	Global minimum
Ackley	$-20 \left( -0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left( \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right)$	$\langle -32.8, 32.8 \rangle$	$(0, \dots, 0)$	0
Griewank	$\sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$	$\langle -600, 600 \rangle$	$(0, \dots, 0)$	0
Rastrigin	$10n + \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i)]$	$\langle -5.12, 5.12 \rangle$	$(0, \dots, 0)$	0
Schwefel	$418.9829n - \sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	$\langle -500, 500 \rangle$	$(0, \dots, 0)$	0
Rotated hyper-ellipsoid	$\sum_{i=1}^n \sum_{j=1}^i x_j^2$	$\langle -65.5, 65.5 \rangle$	$(0, \dots, 0)$	0
Sum squares	$\sum_{i=1}^n i x_i^2$	$\langle -5.12, 5.12 \rangle$	$(0, \dots, 0)$	0
Zakharov	$\sum_{i=1}^n x_i^2 + \left( \sum_{i=1}^n 0.5 i x_i \right)^2 + \left( \sum_{i=1}^n 0.5 i x_i \right)^4$	$\langle -5, 10 \rangle$	$(0, \dots, 0)$	0
Rosenbrock	$\sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$\langle -5, 10 \rangle$	$(0, \dots, 0)$	0
Styblinski-Tang	$\frac{1}{2} \sum_{i=1}^n x_i^4 - 16x_i^2 + 5x_i$	$\langle -5, 5 \rangle$	$(-2.903534, \dots, -2.903534)$	$-39.16599n$

- each cuckoo is interpreted as a point  $\mathbf{x} = (x_0, \dots, x_{n-1})$  in  $n$ -dimensions,
- the nature environment is understood as input domain  $x_i \in \langle a, b \rangle$ ,
- one cuckoo can drop only one single egg in one iteration,
- egg is identified with the cuckoo,
- number of cuckoos in population is constant,
- the host can leave or throw away an egg with a probability  $p_a \in (0, 1)$ . In the case of getting rid of the egg, the cuckoo is forced to find a new place.

The above-mentioned assumptions cause that these birds has a specific environment. Each cuckoo moves in each iteration in search of a better nest for its offspring. The flight is carried out in accordance with the Levy's equations (often called Levy's flight), which is continuous probability distribution defined as

$$L(\mathbf{x}_i, \mu, c) = \sqrt{\frac{c}{2\pi}} \cdot \frac{\exp\left(\frac{c}{2(\mathbf{x} - \mu)}\right)}{(\mathbf{x} - \mu)^{\frac{3}{2}}}, \quad (2)$$

where  $\mu$  is location coefficient and  $c$  is scale parameter. Using Levy's equation, we can move birds on each spatial coordinates as

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha \cdot L(\mathbf{x}_i^t, \mu, c), \quad (3)$$

where  $\alpha$  is the length of Levy's flight,  $t$  is the current iteration.

After the move of cuckoo, the egg is subjected to a certain test depending on whether it will remain in the nest. In case the host notices that he has not his egg, he can throw it away. The probability of discovery is  $p_a$ , and it's modeled as

$$H(\mathbf{x}_i^{t+1}) = \begin{cases} \beta < p_a & \text{drop the egg} \\ \beta \geq p_a & \text{leave the egg} \end{cases}, \quad (4)$$

where  $\beta$  is a random value in range  $\langle 0, 1 \rangle$ . In most cases,  $p_a = 0.5$  what gives the equal likelihood of staying and throwing the egg out of the nest.

### B. Modified version

Proposition is based on modification host decision to prevent the remove eggs that represents good solutions for a given problem. Moreover, every move that cuckoo does is corrected by using a local search method.

1) *Gradient method*: One of the classic local search technique is gradient descent. It is based on the analysis direction of search space by the calculating the negative gradient of function  $f$ . Let assume, that the algorithm starts at position  $\mathbf{x} = (x_0, \dots, x_{n-1})$ . Then, the negative gradient for each spatial coordinates is calculated as

$$-\nabla f_{\mathbf{x}_i} = -\frac{\partial f(\mathbf{x}_0, \dots, \mathbf{x}_{n-1})}{\partial \mathbf{x}_i}, \quad (5)$$

what means the direction of the fastest descent. Then, the value of  $\mathbf{x}_i$  is recalculated as

$$\mathbf{x}_i^t = \mathbf{x}_i^t + \lambda(-\nabla f_{\mathbf{x}_i^t}), \quad (6)$$

where  $\lambda$  is the length of step and  $t$  is the current iteration for these method. After  $T$  iterations, if function value in point  $\mathbf{x}^T$  is better then before local search, it is replaced.

2) *Proposed modification*: A cuckoo moving in search of a nest goes to one point and then the host's decision follows. Let us assume that the egg-laying cuckoo looks for such a place in the nest to minimize the discovery of a new egg. In practice, it can be realized through the local search like method of the smallest gradient described in Algorithm 1.

The second modification is remodeling the host's decision. In the original version depends on the randomness

---

**Algorithm 1** Gradient Descent

---

- 1: Start,
- 2: Define  $f(\cdot)$ , length of the step  $\lambda$ , number of iteration  $T$ ,
- 3: Take point  $\mathbf{x}$ ,
- 4:  $t := 1$ ,
- 5: **while**  $t \leq T$  **do**
- 6:   Calculate  $\mathbf{x}_i^{t+1}$  using (6),
- 7:   **if**  $f(\mathbf{x}_i^{t+1}) \leq f(\mathbf{x}_i^t)$  **then**
- 8:      $\mathbf{x}_i^t = \mathbf{x}_i^{t+1}$ ,
- 9:   **end if**
- 10:    $t++$ ,
- 11: **end while**
- 12: Return  $\mathbf{x}_i^t$ ,
- 13: Stop.

---

of parameter  $\beta$ . This type solution allows to remove well-matched egg in the nest. It is possible to delete the best solution in the whole population, and it leads to a jump in the convergence. Removing the worse solutions by finding the opportunity to the location a better one is justified and therefore remodeling the decision condition is so important – the best solutions should remain, and only the worse ones should be removed. Moreover, the decision must depend on the optimization function. In addition, it would be good if the decision depended also on the quality of the current point with respect to the best in whole population. An example of such a model is presented by the following equation

$$H(\mathbf{x}_i^{t+1}) = \begin{cases} \left| \frac{f(\mathbf{x}_{best}^t)}{f(\mathbf{x}_{current}^t)} \right| < p_a & \text{drop the egg} \\ \left| \frac{f(\mathbf{x}_{best}^t)}{f(\mathbf{x}_{current}^t)} \right| \geq p_a & \text{leave the egg} \end{cases}, \quad (7)$$

where  $\mathbf{x}_{best}^t$  is the best solution in current iteration in whole population.

---

**Algorithm 2** Modified Cuckoo Search Algorithm

---

- 1: Start,
- 2: Define  $p_\alpha$ ,  $c$ ,  $\mu$ , number of individuals  $N$  and number of iteration  $T$ ,
- 3: Define test function  $f(\cdot)$ ,
- 4: Generate  $N$  cuckoos at random,
- 5:  $t:=0$ ,
- 6: **while**  $t \leq T$  **do**
- 7:   Move each cuckoo using Eq. (3),
- 8:   Find the best position in nest using Algorithm 1,
- 9:   Hosts decision is made by Eq. (7),
- 10:    $t++$ ,
- 11: **end while**
- 12: Return the best cuckoo in population,
- 13: Stop.

---

#### IV. EXPERIMENTS

For all test function are presented in Table I. Both versions (classic and modified) were implemented in Mathematica 9.

Calculation were made using two sets of basic parameters which are (the size of population, the number of iterations) and were set as (50, 50) and (100, 100). In addition, the algorithm parameter setting was as follows  $p_a = 0.7$ ,  $c = 0.4$ ,  $\mu = 0.3$  and the step size for gradient descent was 15.

All presented results are the average value from 100 tests calculated as

$$f(\{\mathbf{x}_0, \dots, \mathbf{x}_{99}\}) = \frac{1}{100} \sum_{i=0}^{99} f(\mathbf{x}_i). \quad (8)$$

The obtained averaged solutions are presented in II. From these table, it is clearly to see that for each function, the accuracy is at least slightly better. In the case of function with many local extremes like Rastrigin or Schwefel, the solution is better more then 10% which is a good result considering the possibility of getting stuck. Especially when the algorithm uses local search and this can increase accuracy, but only by finding a better point in a given part of the function.

Not only the accuracy was measured, but also average trajectory in each iteration defined as

$$\frac{1}{100} \sum_{j=0}^{99} \left\| \mathbf{x}_{ideal} - \mathbf{x}_{best}^j \right\|, \quad (9)$$

where  $\mathbf{x}_{ideal}$  is the exactly solution for a given function and  $\mathbf{x}_{best}^j$  is the best point in  $j$ -th test, and  $\|\cdot\|$  is an Euclidean metric defined as the distance between two given points  $\mathbf{p}$  and  $\mathbf{q}$  in  $n$ -dimensions

$$\|\mathbf{p} - \mathbf{q}\| = \sqrt{\sum_{i=0}^n (\mathbf{p}_i - \mathbf{q}_i)^2}. \quad (10)$$

And the last measurement coefficient is rate of convergence in each iteration from the first to the  $t - 1$  calculated as

$$\frac{|f(\mathbf{x}_{k+1}) - f(\mathbf{x}_{ideal})|}{|f(\mathbf{x}_k) - f(\mathbf{x}_{ideal})|}, \quad (11)$$

where  $k$  is the number of the given iteration. All of these values were measured during tests for a population of 100 individuals and 100 iterations. For each iteration, data was saved, then averaged and coefficients calculated in accordance with the above formulas. The illustration of the obtained results are shown in Fig. 1 (for classic version) and in Fig. 2 (for modified version). At first glance, both sets of charts look similar. However, the values on the axes are much smaller in the case of the algorithm with proposed modifications.

In almost all cases, the curves have been slightly smoothed with the introduction of modifications. One of the most interesting cases is the Schwafel function, for which the graph of the average value of the fitness function slowed down to zero. The reason is primarily the landscape (that is, many local minimums) and the use of local search technique which deepened the falling into this minimum. Hence, the function graph is more bulged compared to the original. In other functions, there can be seen improvement.

Table II: Obtained results for all test function in 5 dimensions.

Function	Population iterations	Classic CSA f(x)	Modified CSA f(x)
Ackley	50/50	8.1115809961431699E-3	1.0173805303914699E-3
	100/100	3.1161468790261702E-3	4.25027784882959E-4
Griewank	50/50	1.1104141890427901E-6	1.11533858337188E-9
	100/100	9.7064015180947607E-7	1.6771042340344999E-11
Rastrigin	50/50	1.02208808007731E-3	2.0757391077097499E-5
	100/100	8.5738084933950599E-4	1.6665285423624399E-5
Schwefel	50/50	2.839832E-3	1.62429252212134E-4
	100/100	1.89128E-3	1.4616910059213499E-4
Rotated hyper-ellipsoid	50/50	3.2147278590055301E-6	8.3496728930287895E-7
	100/100	9.774778625110691E-7	1.06247907632132E-7
Sum squares	50/50	3.9473326955707401E-6	3.2406587358828998E-6
	100/100	3.0576751975143799E-6	3.67314536996053E-7
Zakharov	50/50	1.1666325957392999E-5	3.7743509867365198E-6
	100/100	4.8645321838795998E-6	2.7434504482140701E-7
Rosenbrock	50/50	-5.4930904358968698E-2	-6.6770179875013505E-2
	100/100	-8.1294844521185197E-2	-9.8308810919833806E-3
Stybliski-Tang	50/50	-194.11209430931001	-194.2329823892
	100/100	-195.2309123	-195.84287387270001

## V. CONCLUSIONS

In this paper, two modification were presented to improve the performance of Cuckoo Search Algorithm. The first one is the introduction of a local search as finding the best position in the nest to reduce the probability of being detected by the host. The second modification is the remodeling of the host's decision regarding the egg. This is important because in the original algorithm, it was possible to remove the best solutions in a given iteration. And now, the decision depends on the quality of the position relative to the best individual in whole population.

Both versions of CSA were tested in terms of accuracy, quality of adaptation, trajectory, convergence of solution in each iteration. Comparison of results indicates the advantage of modification because the accuracy is more than 10% higher than the original. Moreover, in most cases, convergence was faster except the cases of functions with many local minima. The reason is faster move to the minimum through local search. An important element for future work is to design the exit mechanism from the minimum and to analyze other local search techniques.

## REFERENCES

- [1] S. Anitha and M. M. Metilda. A heuristic approach for observing outlying points in diabetes data set. In *IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials*, pages 199–202. IEEE, 2017.
- [2] Y. Feng, G.-G. Wang, S. Deb, M. Lu, and X.-J. Zhao. Solving 0–1 knapsack problem by a novel binary monarch butterfly optimization. *Neural computing and applications*, 28(7):1619–1634, 2017.
- [3] N. Liu, Q. Chen, J. Liu, X. Lu, P. Li, J. Lei, and J. Zhang. A heuristic operation strategy for commercial building microgrids containing evs and pv system. *IEEE Transactions on Industrial Electronics*, 62(4):2560–2570, 2015.
- [4] M. Ogiolda. The use of clonal selection algorithm for the vehicle routing problem with time windows. *Symposium for Young Scientists in Technology, Engineering and Mathematics*, pages 68–74, 2017.
- [5] E. Okewu, S. Misra, R. Maskeliūnas, R. Damaševičius, and L. Fernandez-Sanz. Optimizing green computing awareness for environmental sustainability and economic security as a stochastic optimization problem. *Sustainability*, 9(10):1857, 2017.
- [6] S. Oreski and G. Oreski. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert systems with applications*, 41(4):2052–2064, 2014.
- [7] J. Peng, Y. Meng, M. Xue, X. Hei, and K. W. Ross. Attacks and defenses in location-based social networks: A heuristic number theory approach. In *International Symposium on Security and Privacy in Social Networks and Big Data*, pages 64–71. IEEE, 2015.
- [8] P. H. V. Penna, A. Subramanian, L. S. Ochi, T. Vidal, and C. Prins. A hybrid heuristic for a broad class of vehicle routing problems with heterogeneous fleet. *Annals of Operations Research*, pages 1–70, 2017.
- [9] D. Połap. Neuro-heuristic voice recognition. In *Federated Conference on Computer Science and Information Systems*, pages 487–490. IEEE, 2016.
- [10] D. Połap et al. Polar bear optimization algorithm: Meta-heuristic with fast population movement and dynamic birth and death mechanism. *Symmetry*, 9(10):203, 2017.
- [11] D. Połap and M. Woźniak. Detection of important features from images using heuristic approach. In *International Conference on Information and Software Technologies*, pages 432–441. Springer, 2017.
- [12] D. Połap, M. Woźniak, C. Napoli, and E. Tramontana. Real-time cloud-based game management system via cuckoo search algorithm. *International Journal of Electronics and Telecommunications*, 61(4):333–338, 2015.
- [13] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, and R. Damaševičius. Is the colony of ants able to recognize graphic objects? In *International Conference on Information and Software Technologies*, pages 376–387. Springer, 2015.
- [14] S. Saremi, S. Mirjalili, and A. Lewis. Grasshopper optimisation algorithm: Theory and application. *Advances in Engineering Software*, 105:30–47, 2017.
- [15] J. T. Starczewski, S. Pabiasz, N. Vladymyrska, A. Marvuglia, C. Napoli, and M. Woźniak. Self organizing maps for 3d face understanding. In *International Conference on Artificial Intelligence and Soft Computing*, pages 210–217. Springer, 2016.
- [16] M. Woźniak, D. Połap, L. Kośmider, and T. Cłapa. Automated fluorescence microscopy image analysis of pseudomonas aeruginosa bacteria in alive and dead stadium. *Engineering Applications of Artificial Intelligence*, 67:100–110, 2018.
- [17] M. Woźniak, D. Połap, L. Kośmider, C. Napoli, and E. Tramontana. A novel approach toward x-ray images classifier. In *IEEE Symposium Series on Computational Intelligence*, pages 1635–1641. IEEE, 2015.
- [18] X.-S. Yang and S. Deb. Cuckoo search via lévy flights. In *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*, pages 210–214. IEEE, 2009.
- [19] K. Z. Zhang, S. J. Zhao, C. M. Cheung, and M. K. Lee. Examining the influence of online reviews on consumers' decision-making: A heuristic-systematic model. *Decision Support Systems*, 67:78–89, 2014.

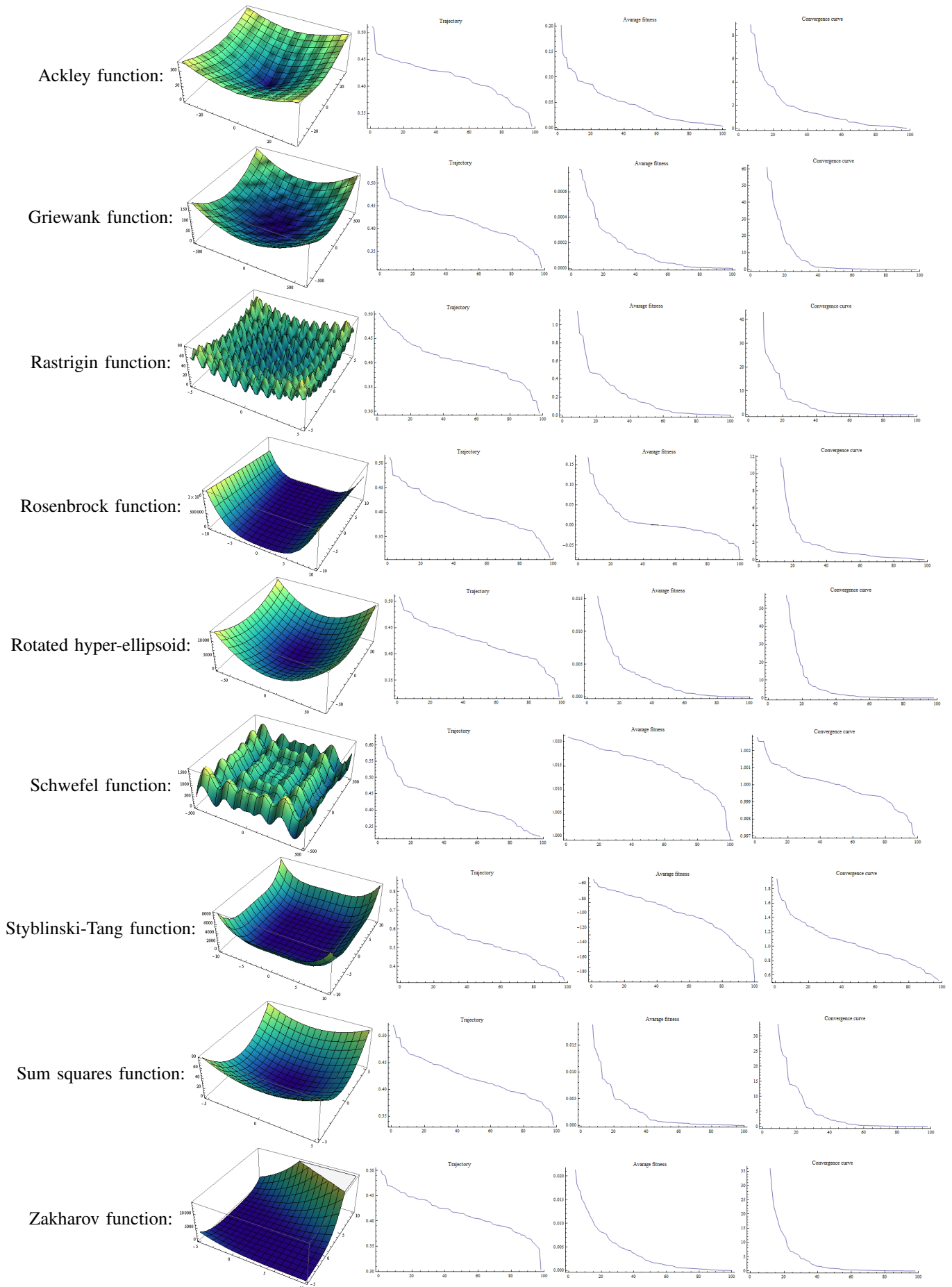


Figure 1: Sample benchmark tests results for all test function obtained by the classic version of CSA. In each row, there are: 3D plot, average trajectory, average fitness function and convergence rate.

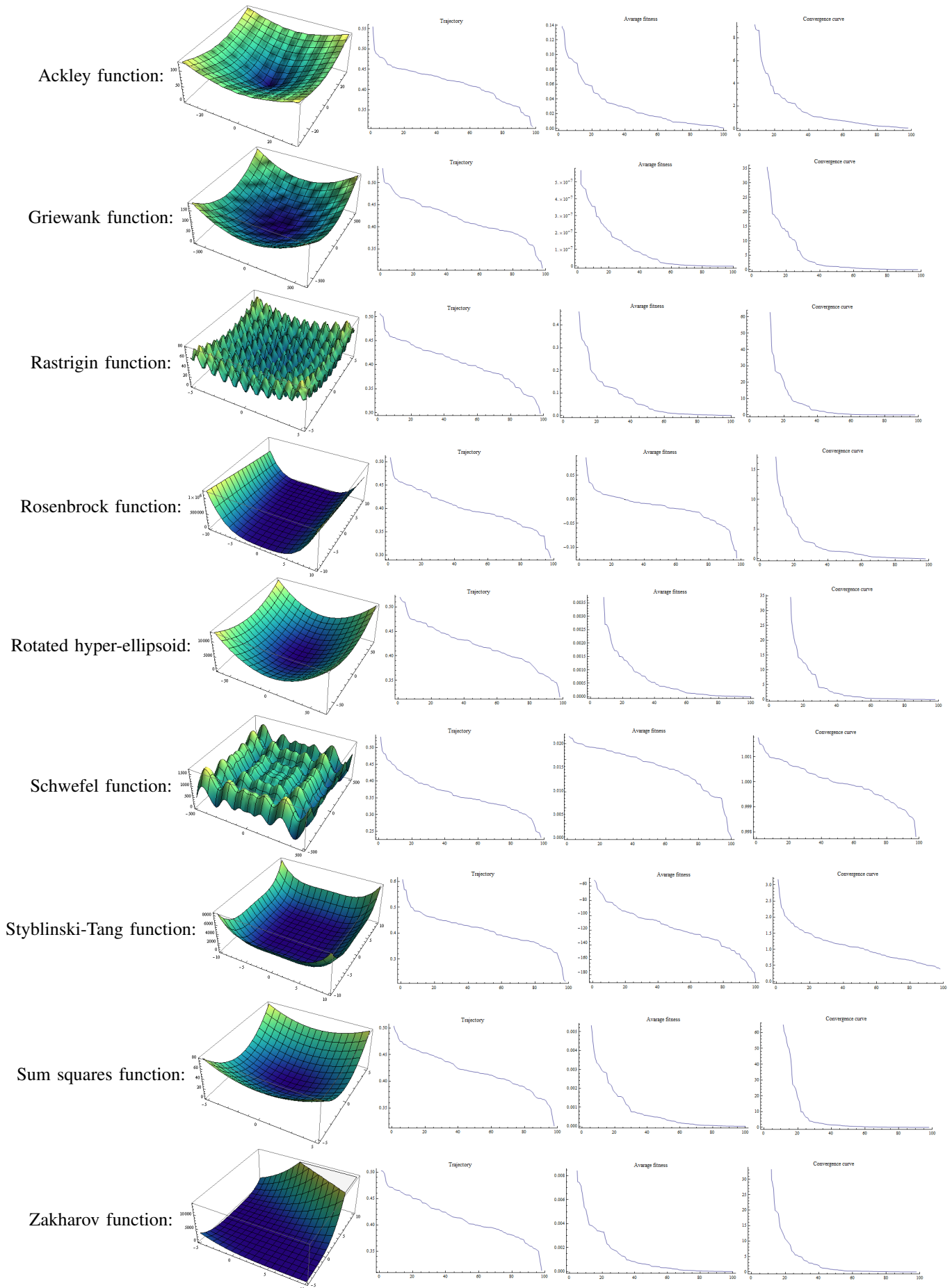


Figure 2: Sample benchmark tests results for all test function<sub>29</sub> obtained by the modified version of CSA. In each row, there are: 3D plot, average trajectory, average fitness function and convergence rate.