IN-TECH Conference

International Conference on Innovative Technologies

IN-TECH2015

DUBROVNIK, CROATIA

09. - 11.09.2015.

Organized by:

World Association for Innovative Technologies

Proceedings

International Conference on Innovative Technologies

IN-TECH 2015

Dubrovnik

Proceedings

IN-TECH 2015

Proceedings of International Conference on Innovative Technologies

Editors:

- Zlatan Car – Croatia
- Jan Kudláček – Czech Republic

IN-TECH 2015 Organization committee:

- Zlatan Car – Croatia
- Jan Kudláček – Czech Republic
- Tomaž Pepelnjak – Slovenia
- Leon Šikulec – Croatia
- Karel Vojkovsky – Czech Republic
- Petr Drašnar – Czech Republic
- Petr Zikmund – Czech Republic
- Michal Pakosta – Czech Republic

Publisher: Faculty of Engineering University of Rijeka

Printed by: Tiskara Sušak, Rijeka, Croatia

Printed in 200 copies.

IN-TECH 2015 International Conference on Innovative Technologies runs from 08.09.2015. to 12.09.2015. in Dubrovnik, Croatia.

E-mail: info@in-tech.info
URL: http://www.in-tech.info

ISSN 1849-0662

APPLICATION OF TEST FUNCTIONS FOR THE EVALUATION OF METAHEURISTIC ALGORITHMS

Károly Jármai[1], Gábor Zoltán Marcsák[2], Csaba Barcsák[3]

[1] Professor, University of Miskolc, H-3515 Miskolc, Egyetemváros, Hungary

[2] application developer, DACHS Hungary Ltd., [3] software engineer, LogMeIn Ltd., Hungary

Keywords: structural optimization, metaheuristic algorithm, test function, benchmark problem

Abstract. In this paper the analysis and comparison of very recent heuristic methods are made. It is clear, that any two optimization algorithms can be equivalent when their performance is averaged across all possible problems (no free lunch theorem). The authors acknowledge that their paper is unlikely to answer what is the best algorithm today, however, the paper can attempt to analyse what are the strengths and weaknesses of the newly proposed heuristic methods. They have made comparisons using special test functions, where without changing the position of the optimum, they can make the objective function to be noisier.

Introduction

The advantage of heuristic algorithms is that they can find approximate solutions, even when the search space is excessively huge. However, finding the global optimum cannot be guaranteed, since they don't evaluate all feasible solutions. A good heuristic algorithm has to maintain balance between local search and global search. On one hand, it has to explore the entire search space properly, on the other hand search around the current best positions efficiently. In other words, quickly find regions with quality solutions, and don't waste too much time in low quality areas. Most of the time, heuristic algorithms have stochastic behaviour. Ideally, the final solutions, through slightly different, will converge to the optimal solution of the given problem.

Nowadays a lot of nature inspired heuristic algorithms emerge. We benchmarked sixteen optimization techniques, like evolutionary (Differential Evolution, Cultural Algorithm, Memetic Algorithm, Genetic Algorithm, Cuckoo search), physical (Simulated Annealing, Harmony Search, Cross entropy), biological (Artificial Immune Network) and swarm intelligence (Bacterial Foraging, Bees Algorithm, Krill Herd, Particle Swarm, Bat Algorithm, Firefly Algorithm) optimization methods.

Name	Definition	Search range and global optimum
Ackley's function (F1)	$f(x) = -20 \times \exp(-0.2 \times \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i^2)} - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + \exp(1)$	$-32.768 \leq x_i \leq 32.768$ $x_i = 0, i = 1, \dots, n$ f(x)=0
De Jong's function (F2)	$f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \leq x_i \leq 5.12$ $x_i = 0, i = 1, \dots, n$ f(x)=0
Drop-Wave function (F3)	$f(x) = -\frac{1 + \cos(12 \times \sqrt{x_1^2 + x_2^2}}{\frac{1}{2}(x_1^2 + x_2^2) + 2} + 1$	$-5.12 \leq x_i \leq 5.12$ $x_i = 0, i = 1, 2$ f(x)=0 (min)
Easom's function (F4)	$f(x) = -\cos(x_1) \times \cos(x_2) \times \exp(-(x_1 - \pi)^2 - (x_2 - \pi)^2) + 1$	$-10 \leq x_i \leq 10$ $x_i = \pi, i = 1, 2$ f(x)=0
Griewangk's function (F5)	$f(x) = \frac{1}{4000} \times \sum_{i=1}^{n} x_i^2 - \prod_{\mathrm{i}=1}^{\mathrm{n}} \cos\left(\frac{x_\mathrm{i}}{\sqrt{\mathrm{i}}}\right) + 1$	$-600 \leq x_i \leq 600$ $x_i = 0, i = 1, \dots, n$ f(x)=0
Matyas's function (F6)	$f(x) = 0.26 \times (x_1^2 + x_2^2) - 0.48 \times x_1 \times x_2$	$-10 \leq x_i \leq 10$ $x_i = 0, i = 1, 2$ f(x)=0
Rastrigin's function (F7)	$f(x) = 10 \times n + \sum_{i=1}^{n}[x_i^2 - 10 \times \cos(2\pi x_i)]$	$-5.12 \leq x_i \leq 5.12$ $x_i = 0, i = 1, \dots, n$ f(x)=0
Rosenbrock's valley (F8)	$f(x) = \sum_{i=1}^{n-1}[100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$	$-2.048 \leq x_i \leq 2.048$ $x_i = 1, i = 1, \dots, n$ f(x)=0
Schaffer's N. 2. function (F9)	$f(x) = 0.5 + \frac{sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001 \times (x_1^2 + x_2^2)]^2}$	$-10 \leq x_i \leq 10$ $x_i = 0, i = 1, 2$ f(x)=0
Three-hump camelback (F10)	$f(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1 x_2 + x_2^2$	$-2.048 \leq x_i \leq 2.048$ $x_i = 0, i = 1, 2$ f(x)=0

A lot of mathematical test functions can be found in the literature, for example in the paper of Mologa and Smutnicki [1]. The complexity of test functions is determined by the number of variables and the number and distribution of local extremes. Continuous test functions have been studied with two variables, since those problems can be plotted as 3D surfaces. Table 1 summarizes the used ten test functions, which are in alphabetical order. The Ackley's function has a nearly flat outer region, and a large valley at its centre. This widely used multimodal test function can easily trap metaheuristic algorithms at one of its local optima. De Jong's function is a very simple, convex, unimodal benchmark problem. The Drop-Wave function is very complex, with expanding ripples, like when an object is dropped into liquid surface. Easom's function is unimodal like De Jong's, however more complicated, because the global optimum is relatively small, compared to the search space. Griewangk's function looks similar to De Jong's either, but it has a rugged surface with many regularly distributed local optima. Matyas's function is a plate shaped problem, it doesn't have any local extremes, only the global one, which is relatively easy to find. However, convergence to the global optima is difficult, so that is a great number of benchmark problems to measure the accuracy and convergence rate of search algorithms. Rastrigin's function, also known as egg holder, is a widely used, highly multimodal problem with regularly distributed local extremes. Rosenbrock's valley is unimodal, the global minimum can be found in a narrow, parabolic valley. Schaffer's second function is an extremely noisy optimization problem, with a lot of local optima very close to each other. Last but not least, Three-hump camelback looks like Rosenbrock's valley, however it has two local extremes.

Main characteristics of metaheuristic algorithms

As mentioned before, the biggest advantage of metaheuristic algorithms is that they can find approximate solutions, even when the search space is excessively huge. However, finding the global optimum cannot be guaranteed, since they don't evaluate all feasible solutions. With the utilization of metaheuristic algorithms, one obtains computing speed, but the accuracy can be the price to pay for it. A good metaheuristic algorithm has to maintain balance between local search and global search. On one hand, it has to explore the entire search space properly, on the other hand search near the current best positions should be efficient. In other words, it quickly finds regions with quality solutions, and doesn't waste too much time in low quality areas. Most of the time, heuristic algorithms have stochastic behaviour. Ideally, the final solutions, though slightly different, will converge to the optimal solution of the given problem. However, the way how metaheuristic algorithms get to the solution is always a bit different because of the stochastic factor. A lot of nature inspired metaheuristic algorithms emerge based on the behaviour of biological and physical systems.

Then optimization techniques have been benchmarked like evolutionary (Cultural Algorithm, Differential Evolution, Memetic Algorithm), physical (Harmony Search, Simulated Annealing, Cross-Entropy Method), and swarm intelligence (Bacterial Foraging, Bat Algorithm, Bees Algorithm, Cuckoo Search, Firefly, Particle Swarm and Multi-Swarm Optimization) and other methods like Nelder-Mead and Random Search optimization methods. The algorithms' source codes are available on the Internet. The algorithms search for the global minimum of the Rastrigin's function in the source code Marcsák and Jármai [2].

Bacterial Foraging Optimization Algorithm (BFOA) was first described by Liu and Passino [3]. It's a relatively new swarm intelligence search algorithm. These techniques use the collective intelligence of numerous homogenous individuals. In principle, an individual entity may not be able to solve a problem on its own. However, if a large number of individuals form a group, the group's collective intelligence may be enough to solve the task. Bacterial Foraging is based on the foraging and reproduction behaviour of E. Coli bacteria colonies. The chemotaxis movement of the group tends toward precious nutrients (global optima), while avoiding dangerous places (local optima) in the environment.

Bat Algorithm (BATA) is a swarm intelligence metaheuristic method published by Yang [4], based on the echolocation behaviour of bats. Echolocation allows bats to find and identify their prey (global optimum), and avoid obstacles (local optima), even in complete darkness. Echolocation works like a sonar, as bats emit a sound pulse and listen for the echo that bounces back from the surrounding environment. During the search, frequency, loudness and emission rate of the individual's sound changes regarding the position of the prey, which informs other bats. The loudness of the emitted sound decreases as the bat approaches the prey, while the rate of pulse emission increases. Bat Algorithm combines the advantages of existing swarm intelligence methods, such as Particle Swarm Optimization and Firefly Algorithm.

Bees Algorithm (BA) was published by Pham et al. [5]. Primarily it was developed to search for the global optima of continuous mathematical functions. It belongs to the field of swarm intelligence procedures. The Bees Algorithm, as the name suggests, was inspired by the foraging behaviour of honey bees. Bee colonies send scout bees to explore the environment, and find areas rich of nectar (global or local optima). The scout bees then return to the hive and inform the worker bees about the position and quality of food sources. The number of worker bees sent to each food source depends on these characteristics. The scout bees continually search for good sites, while worker bees accurately explore the already found spots. Generally speaking, the scout bees are responsible for global search, while worker bees provide good local search. The algorithm is similar to Ant Colony Optimization and Particle Swarm optimization, however the rank system of bees makes it unique. The Bees Algorithm is capable to solve both continuous and combinatorial optimization problems.

Cross-Entropy Method (CEM) is a probabilistic optimization algorithm developed by Rubinstein [6]. The name of the technique comes from the Kullback-Leibler cross-entropy divergence, as a measure of closeness between two probability distributions. Cross-Entropy Method is an adaptive importance estimation technique for rare-event probabilities in discrete event simulation systems. Optimization problems can be described as rare-event systems, because the probability of locating an optimal solution by using random search is a rare-event probability. The method changes the sampling distribution of random search, so that the rare-event (finding the optima) is more likely to occur.

Cuckoo Search Algorithm (CS) was developed by Yang and Deb [7]. The algorithm was inspired by the obligate brood parasitism of certain cuckoo species, which means they lay their eggs in the nests of other host species. Sometimes if the host birds discover the eggs are not their own, they throw the alien eggs away, or simply abandon the current nest and build a new elsewhere. The algorithm randomly distributes a fixed number of nests through the search space. The cuckoos lay down their eggs, each cuckoo lays one egg at a time in a randomly chosen nest. If the egg is discovered by the host bird, it can either throw the egg away, or abandon the current nest and build a completely new elsewhere. Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim of the algorithm is to find new, potentially better solutions (global optima) to replace the worse (local optima). The best nests with high quality of eggs will be members of the next generation.

Cultural Algorithm (CA) was described by Reynolds [8]. This evolutionary algorithm simulates the cultural evolution of human society. Culture includes habits, beliefs and morals of a member of the society. Culture can interact with the environment via positive or negative feedback cycles. As the evolutionary process goes on, individuals gain information about the search space, which is communicated to other individuals in the population. This creates a knowledge base that stores positive feedback about useful areas of the environment (global optimum), as well as potentially hazardous areas (local optima). This cultural knowledge base is expanded and exploited through generations as situations change.

Differential Evolution (DE) algorithm was developed by Storn and Price [9], and belongs to the field of evolutionary algorithms. Differential Evolution is mainly based on Darwin's Theory of Evolution, because its main principle is natural selection. The algorithm maintains a population of candidate solutions with recombination, evaluation and selection as generations unfold. The recombination creates a new candidate solution based on the weighted difference between two randomly selected population members added to a third population member.

Firefly Algorithm (FF) is a nature-inspired metaheuristic optimization algorithm for multimodal optimization developed by Yang [9]. The algorithm was inspired by the flashing behaviour of fireflies, in order to communicate with other fireflies. The flashing light is produced by a biological process called bioluminescence, and can attract mating partners as well as potential preys. It can also serve as a protective warning mechanism. The rate of flashing and the light intensity are very important characteristics of the communication. The flashing light can be associated with the objective function to be optimized. As a firefly gets closer to a good solution the more light it will emit. The less bright fireflies will move toward the brighter ones. Attractiveness is proportional to the brightness and it decreases as the fireflies distance increases. If there is no brighter one than a particular firefly it will move randomly. Firefly Algorithm has adjustable visibility and more versatile on attractiveness variations than other techniques, like Particle Swarm Optimization.

Harmony Search (HS) was published by Geem et al. [10]. It was inspired by the improvisation of Jazz musicians. When they start a musical performance, they adapt their music to the band, creating musical harmony. If a false sound occurs, each individual of the band makes modifications to improve their performance. The musicians seek harmony through small variations and improvisation, and the harmony is taken as a complete candidate solution. The audiences' aesthetic appreciation of the harmony, represent the cost function. The algorithm has some similarities to Cultural Algorithm, since the candidate solution's components are stochastically created either directly from the memory of high-quality solutions, adjusted from the memory of high-quality solutions, or assigned randomly Brownlee J [11].

Memetic Algorihm (MA) was developed by Moscato (1989). The algorithm simulates the creation and inheritance of cultural information among individuals. Meme is the basic unit of cultural information (an idea, discovery, etc.), which name derives from the biological term gene. Universal Darwinism is the generalization of genes beyond biological-based systems to any system, where discrete units of information can be distributed, and subjected to evolutionary changes. The objective of the algorithm is to make a population based global search, while the individuals explore the promising areas with a local search. Balance is crucial between global and local search mechanisms to ensure the algorithm doesn't get struck at a local optimum, while it spares computational resources. Meme is information about the search, shared between individuals through generations, which influence the evolutionary processes. Memetic Algorithm is a duality of genetic and cultural evolutionary methods.

Nelder-Mead (NM) algorithm was named after its creators, as Nelder and Mead (1965) proposed this metaheuristic algorithm. In the literature (McCaffrey 2014) the method is often referred as Amoeba Method. Basically Nelder-Mead is a simplex search algorithm, commonly used for nonlinear optimization problems. The algorithm creates random candidate solutions, and each solution has an associated fitness function value. The candidates are ordered by their fitness, and at each generation, the algorithm attempts to replace the worst solution with a better one. The better solution is chosen from among three candidates, the reflected point, the expanded point and the contracted point. All of these points lie along a line from the worst point through the centroid. The centroid is in the middle of all points except the worst point. If neither the points are better than the current worst solution, the amoeba moves all points, except for the best point, halfway towards the best point.

Particle Swarm Optimization (PSO) swarm intelligence metaheuristic algorithm was developed by Eberhart and Kennedy (1995), Kennedy (2010). Particle Swarm's operation was inspired by the foraging movement of bird and fish swarms. The particles of the swarm move around the search space towards historically good areas. A particle's new position is influenced by the best known position of their own, also by the best known position of the whole swarm. The mathematical formulae of the movement determine the new velocity and position of particles, guiding the swarm towards the global optima. The process is repeated in each generation, while stochastic factors also affect the particles' movement.

Multi-Swarm Optimization (MSO) is a variant of Particle Swarm Optimization (PSO). Instead of using one swarm, MSO uses a user-defined number of swarms to locate the global optima (Zhao et al. 2008). The algorithm is especially good for multi-modal optimization problems, where numerous local optima exist. Multi-Swarm Optimization is a new approach to improve the balance between global and local search.

Random Search (RS) algorithm, as the name suggests, is a simple random search algorithm. It takes any position in the search space with equal probability (Brooks 1958). The new solutions are always independent from the previous ones. Random search provides a candidate solution construction and evaluation routine.

Simulated Annealing (SA) method was described by Kirkpatrick (1983). The operation of the algorithm is based on a physical phenomenon. In metallurgy, certain materials gain beneficial properties when heated and then cooled under controlled conditions. The materials' crystal structure is transformed during the process, because the particles take more favourable positions. The metaheuristic algorithm emulates this process to search for better solutions to a given problem. Each solution of the algorithm represents a particle, the acceptance of new position is controlled by the Metropolis-Hastings Monte Carlo algorithm. As the system is cooled, the acceptance criteria of samples are narrowed to focus on improving movements. Theoretically, if the cooling process is long enough, the system always converge to the global optima. The continuous version of the algorithm was developed by Corana et al. (1987).

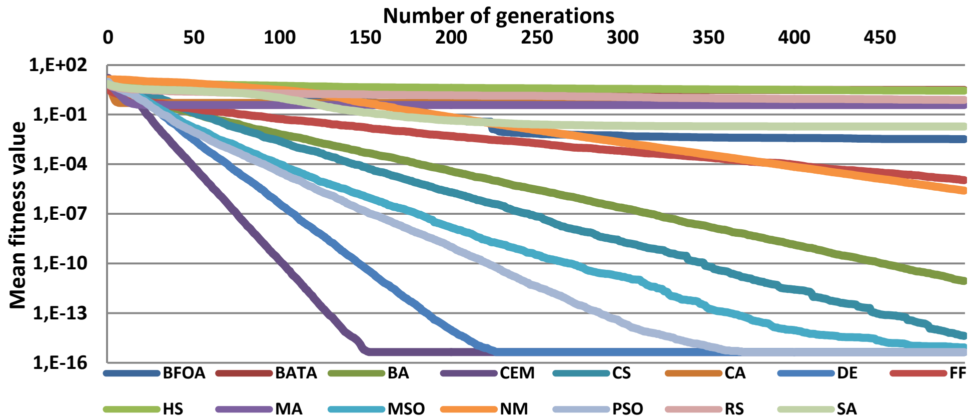
Table 1. Mean normalized optimization results for thirteen benchmark functions

	BFOA	BATA	BA	CEM	CS	CA	DE	FF	HS	MA	MSO	NM	PSO	RS	SA
F1	0,11	100,00	**0**	**0**	**0**	16,45	**0**	**0**	91,79	12,49	**0**	**0**	**0**	25,28	0,63
F2	0,01	**0**	**0**	**0**	**0**	100,00	**0**	**0**	48,51	1,57	**0**	**0**	**0**	5,36	**0**
F3	2,82	100,00	**0**	14,67	**0**	47,96	0,17	0,02	76,81	13,93	**0**	63,26	**0**	27,15	50,81
F4	**0**	67,91	3,05	**0**	**0**	11,78	75,26	18,81	62,98	60,31	**0**	100,00	**0**	1,68	3,33
F5	100,00	31,93	**0**	0,43	**0**	4,38	0,01	0,03	40,28	1,90	**0**	3,55	0,11	8,03	28,69
F6	**0**	**0**	**0**	**0**	**0**	13,84	2,50	**0**	100,00	35,87	**0**	**0**	**0**	0,86	0,03
F7	0,24	49,39	**0**	2,28	**0**	82,37	**0**	0,17	43,75	22,23	**0**	100,00	**0**	19,63	**0**
F8	0,03	**0**	4,65	**0**	0,24	100,00	0,11	**0**	86,60	32,11	**0**	**0**	**0**	2,63	**0**
F9	100,00	65,45	**0**	**0**	**0**	21,40	0,01	**0**	48,21	7,25	**0**	76,73	**0**	3,92	23,27

F10	0,02	100,00	**0**	**0**	**0**	16,45	**0**	**0**	2,74	1,13	**0**	**0**	**0**	1,53	**0**
Σ	**2**	**3**	**8**	**7**	**9**	**0**	**4**	**6**	**0**	**0**	**10**	**5**	**9**	**0**	**4**

Convergence plots have been used in Table 2 to measure the convergence rate of search algorithms. The numbers are normalized between 0 and 100, which means that 0 is the best, 100 is the worst solution. Convergence rate shows how quickly the metaheuristic algorithms can find the optimum. The data points are the best fitness found in each iteration, averaged for 100 Monte Carlo simulations.

Table 2. Convergence plots for Ackley's (F1) function



Ten search algorithms have been used for the n benchmark problems. The test functions had quite diverse characteristics, like unimodal and multimodal problems with varying number and distribution of local extremes. After the examination of the significant amount of statistical data, the algorithms' overall performance became more clear. If one takes a look at Table 1, one can see the superiority of swarm intelligence methods. Multi-Swarm Optimization (MSO) always found the global optima, while Cuckoo Search (CS) and Bees Algorithm (BA) almost always found the global optima, even in case of very difficult and noisy functions. Additionally, Cross-Entropy Method and Simulated Annealing (SA) worked well, however sometimes they trapped at local optima. To improve the performance of slowly converging methods (HS, SA), one should increase the number of iterations. The rate of convergence could be observed well on the convergence plots of Table 2. The best algorithms' rate of convergence is relatively fast, and this characteristic proved to be crucial for success. Furthermore, the convergence plots showed the weighting functions really affect the difficulty of complex functions.

Conclusion

The field of numerical optimization is an always evolving scientific field. Several new evolutionary optimization techniques appeared lately. A software solution have been created, which provides practically endless possibilities to create arbitrarily difficult complex test functions. Fifteen optimization algorithms with ten test functions have been benchmarked. A comparison has been made with Brute Force method. Both the test functions design show that the evolutionary optimization techniques are very efficient tools of the design. Multi-Swarm Optimization (MSO) always found the global optima, while Cuckoo Search (CS) and Bees Algorithm (BA) almost always found the global optima, even in case of very difficult and noisy functions. Additionally, Cross-Entropy Method and Simulated Annealing (SA) worked well, however sometimes they trapped at local optima. In the future, our plane is to create more difficult, self-made test problems, and benchmark further metaheuristic techniques. Based on the benchmark result novel, more efficient hybrid metaheuristic algorithms have been created, which could be utilized in real-life structural and logistics optimization problems.

Acknowledgment

The research was supported by the Hungarian Scientific Research Fund OTKA T 109860 project and was partially carried out in the framework of the Center of Excellence of Innovative Engineering Design and Technologies at the University of Miskolc.

References

[1] Mologa M, Smutnicki C.: Test functions for optimization needs, pp. 1-10, http://www.bioinformaticslaboratory.nl/twikidata/pub/Education/NBICResearchSchool/Optimization/VanKampen/BackgroundInformation/TestFunctions-Optimization.pdf (access July 29, 2014)
[2] Marcsák G, Jármai K.: C# source code of metaheuristic algorithms, https://drive.google.com/folderview?id=0BxE6yHbGFZBAOHBpN2VIV08ySOk&usp=sharing (access May 2, 2015)
[3] Liu Y, Passino KM (2002) Biomimicry of social foraging bacteria for distributed optimization: Models, principles, and emergent behaviours, Journal of Optimization Theory and Applications, pp. 603–628.
[4] Yang XS (2010) A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization (NISCO 2010), (Eds. J. R. Gonzalez et al.), Studies in Computational Intelligence, Springer Berlin, Springer, pp. 65-74
[5] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S, Zaidi M (2005) The bees algorithm. Technical report, Manufacturing Engineering Centre, Cardiff University
[6] Rubinstein RY (1997) Optimization of computer simulation models with rare events. European Journal of Operations Research, 99, pp. 89–112.
[7] Yang XS, Deb S (2009) Cuckoo search via L´evy flights, Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. IEEE Publications, pp. 210-214.
[8] Reynolds RG (1994) An introduction to cultural algorithms. In Proceedings of the 3rd Annual Conference on Evolutionary Programming, World Scientific Publishing, pp. 131–139.

[9] Yang XS (2009) Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications, SAGA 2009. Lecture Notes in Computer Sciences 5792. pp. 169-178.
[10] Geem ZW, Kim JH, Loganathan GV (2001) A new metaheuristic optimization algorithm: Harmony search. Simulation, 76(2), pp. 60–68.
[11] Brownlee J (2011) Clever Algorithms: Nature-Inspired Programming Recipes, Lulu, pp. 27-336, ISBN: 978-1-4467-8506-5