©2019 Published in 7th International Symposium on Innovative Technologies in Engineering and Science 22-24 November 2019 (ISITES2019 SanliUrfa - Turkey)
https://doi.org/10.33793/acperpro.02.03.63

A Metaheuristic-Based Tool for Function Minimization

[1]Yigit Çağatay Kuyu and *[1]Fahri Vatansever
*[1]Faculty of Engineering, Electrical-Electronics Engineering Dept., Bursa Uludag University, Turkey

Abstract

During the last decade, metaheuristic algorithms have occupied an important place in the field of optimization. Function minimization is of importance to researchers since many real-world problems can be modeled mathematically and be solved effectively through metaheuristic algorithms. Due to the growing scientific interest in the field of optimization and the good performances shown by the algorithms on function minimization, the practical and quick implementation concept is necessary to select the most appropriate algorithms on function minimization, and to assist researchers in analyzing the performance of the algorithms. In this study, a tool is developed to minimize user-defined functions in a specified range according to the chosen metaheuristic algorithms, which allows analyzing the algorithms in the general experimental environment. The tool, which has a user-friendly interface, can provide single and comparative solutions by simultaneously executing the algorithms. Each solution and computational time obtained by the algorithms is given numerically, and the convergence behavior of the algorithms is shown graphically in the tool interface. Minimization of functions can be made fast, easily and effectively through the developed tool.

Key words: Function minimization, metaheuristic algorithms, software tool.

1. Introduction

In recent years, there has been an increase interest in development and application of optimization algorithms, especially in the field of metaheuristic algorithms, which enables researchers to focus on in this field for solving minimization problems. Traditional methods can get stuck in local minimum points in function minimization, may require derivative information and also convergence to optimal solutions may depend on the chosen initial points [1-2]. Alternatively, metaheuristic algorithms offer a derivative-free approach to guide the search process of near-optimal solutions, and can be applied on a wide range of problems [2-3]. In addition, these algorithms are easy to use on different problems without the need of expert knowledge, which make these algorithms more robust and effective than the traditional methods. Metaheuristic algorithms can be inspired by the some aspects of natural evolution, physics, ecology and so forth in their development process. These algorithms perform different stochastic search strategies to achieve their solutions. Generally, in the first step, candidate solutions to a problem, frequently called a population, are created randomly, and then the algorithm enters into a loop. In the second step, each individual of the population is a candidate solution and evaluated according to the given objective function, which represents a fitness value. New solutions are then created inside its population. Fitness values of the individuals are used for how well they address the problem. The individuals of the population can also share solution information with the other individuals to reach more promising regions of the search space. The improvement of the solutions during the searching process depends on the algorithm's own special operations in

*Corresponding author: Address: Faculty of Engineering, Department of Electrical-Electronics Engineering, Bursa Uludag University, 16059, Bursa, TURKEY. E-mail address: fahriv@uludag.edu.tr, Phone: +902242940905

the search space. Finally, a selection criterion (number of iterations, tolerance, number of function evaluations, etc.) is applied to stop the searching process of an optimal solution and the final result found by the algorithm is given as an optimal solution of the problem.

Real-world problems are first modeled mathematically and then transferred to the computer environment. Generally, errors are minimized in these problems hence the problem of minimization of functions becomes important. Besides, there is still the need of the tools to minimize functions for general purpose for researchers to quickly and properly analyze the performance of the algorithms since analyzing the results of mathematical functions may help researchers in designing algorithms for real-world applications. Metaheuristic algorithms are suitable for solving these problems due to the aforementioned advantages. They can achieve successful solutions on one type of problem but may not get satisfactory results on the other type of problem. Therefore, the performance analyses of the algorithms are necessary to understand which algorithm is better on which type of the problem at a reasonable process time.

The motivation of this study is to develop a tool for minimizing functions according to the user-defined parameters, function and chosen metaheuristic algorithm(s), which would be beneficial to find a solution to the different type of mathematical functions quickly and effectively. The developed tool has an ability to show the results to the user through its user-friendly interface numerically and graphically. The tool includes six metaheuristic algorithms, which are artificial bee colony (ABC) [4], backtracking search optimization (BS) [5], cuckoo search (CS) [6], firefly (FF) [7], harmony search (HS) [8] and vortex search (VS) [9] algorithms, and is able to provide the analyses of the algorithms single or comparatively. The obtained optimal solutions by the algorithms are listed in the tool interface. The convergence graphs of the algorithms to solutions and the computational time of each algorithm used are also shown through the interface, which enables the user to easily compare several metaheuristic algorithms at one time. The usage of the tool can be explained step by step:

- The function to be minimized and boundaries of the search space are written in the tool interface,
- Two algorithm-specific parameters as iteration number and population size are defined,
- The algorithm to be used and type of analyze are chosen by the user.

After the aforementioned processes are completed, the tool will show the optimal solutions found by the chosen algorithm(s), convergence graph and process time of the selected algorithm(s).

The rest of this paper is organized as follows: The brief descriptions of the metaheuristic algorithms used are given in Section 2. The main screen of the tool interface and applications of the tool are presented in Section 3. Finally, the study conclusions are detailed in Section 4.

2. Metaheuristic Algorithms

Metaheuristic algorithms are used on a broad range of different disciplines, such as engineering, scientific, education, etc. The process mechanisms of the algorithms used in this study can be briefly described in Table 1[4-9].

Table 1. The process mechanisms of the algorithms

ABC	BS	CS	FF
1. Define number of colony size and limit value. 2. Generate initial food source positions. 3. Calculate initial nectar amounts. 4. Is the termination criteria satisfied? a. If satisfied: - Final food positions are best food positions. b. (Else) If not satisfied: - Determine the new food positions. - Calculate the nectar amounts and apply selection process. - Memorize the position of best food source so far. - Go to step 3.	1. Define the population size and mix rate. 2. Generate the initial population randomly. 3. Evaluate the fitness of initial population. 4. Is the termination criteria satisfied? a. If satisfied: - Choose the best individual with respect to fitness values as a solution. b. (Else) If not satisfied: - Apply selection-1, mutation and crossover to get trial population. - Evaluate the fitness of trial population. - Apply selection-2 process. - Go to step 4.	1. Define number of nests and probability of discovery egg by the host bird. 2. Generate initial population of host nests. 3. Calculate Cuckoo solutions. 4. Apply replacing process and determine Cuckoo societies. 5. Is the termination criteria satisfied? a. If satisfied: - The best Cuckoo solution is found. b. (Else) If not satisfied: - Go to step 3.	1. Define number of fireflies, largest degree of attraction, degree of light attenuation and step factor. 2. Generate initial population randomly. 3. Calculate the relative brightness and attraction between fireflies. 4. Is the termination criteria satisfied? c. If satisfied: - Obtained minimum location from best firefly d. (Else) If not satisfied: - Test brightness for moving and update position. - Go to step 3.

HS	VS
1. Define harmony memory size, harmony memory consideration rate and pitch adjustment rate. 2. Initialize harmony memory randomly. 3. Calculate harmony memory solution. 4. Is the termination criteria satisfied? a. If satisfied: - Best harmony in the harmony memory is the solution. b. (Else) If not satisfied: - Improve new harmony. - Update harmony memory. - Go to step 3.	1. Define the number of neighborhood solutions. 2. Generate candidate solutions. 3. Evaluate the fitness of each candidate solution and keep the best solution so far. 4. Is the termination criteria satisfied? a. If satisfied: - Choose the best solution so far. b. (Else) If not satisfied: - Decrease radius. - Update the candidate solutions. - Go to step 3.

3. Developed Tool and Its Applications

The tool was developed under the MATLAB platform [10]. The main screen of the tool is given in Fig 1. In this screen, an equation of function, search space boundaries, population size and maximum iteration number as a stopping criterion are entered, and the user can choose analysis type and the algorithm(s) to be used (totally six metaheuristic algorithms) in the tool interface in accordance with his/her preferences. Then the tool performs analyses on the given function dependent on selected algorithm(s). After finishing the analyzing process, the variable(s) and computational time obtained by the algorithm(s) are listed, and the convergence graph of the selected algorithm(s) are given in the tool interface.

In the first application, which can be seen in Fig. 2, according to the results of input equation [9]:

$$(1.5 - X_1 + X_1X_2)^2 - (2.25 - X_1 + X_1X_2^2)^2 + (2.625 - X_1 + X_1X_2^3)^2 \quad (1)$$

with BS algorithm.

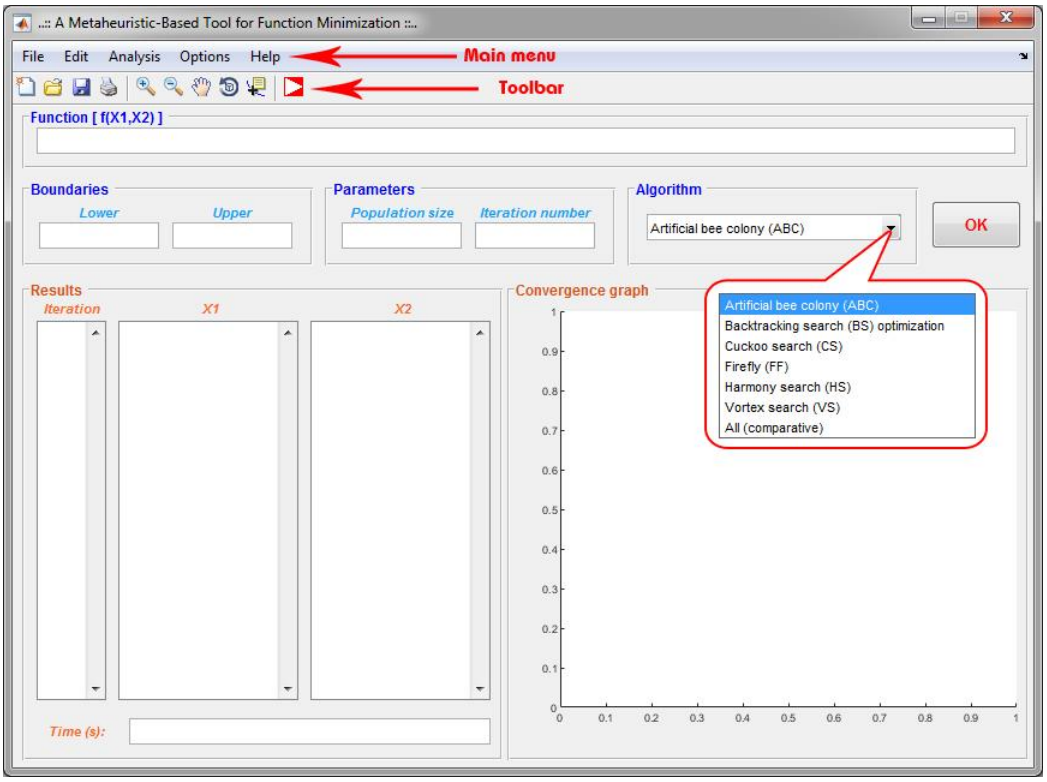

Figure 1. The main screen of designed tool

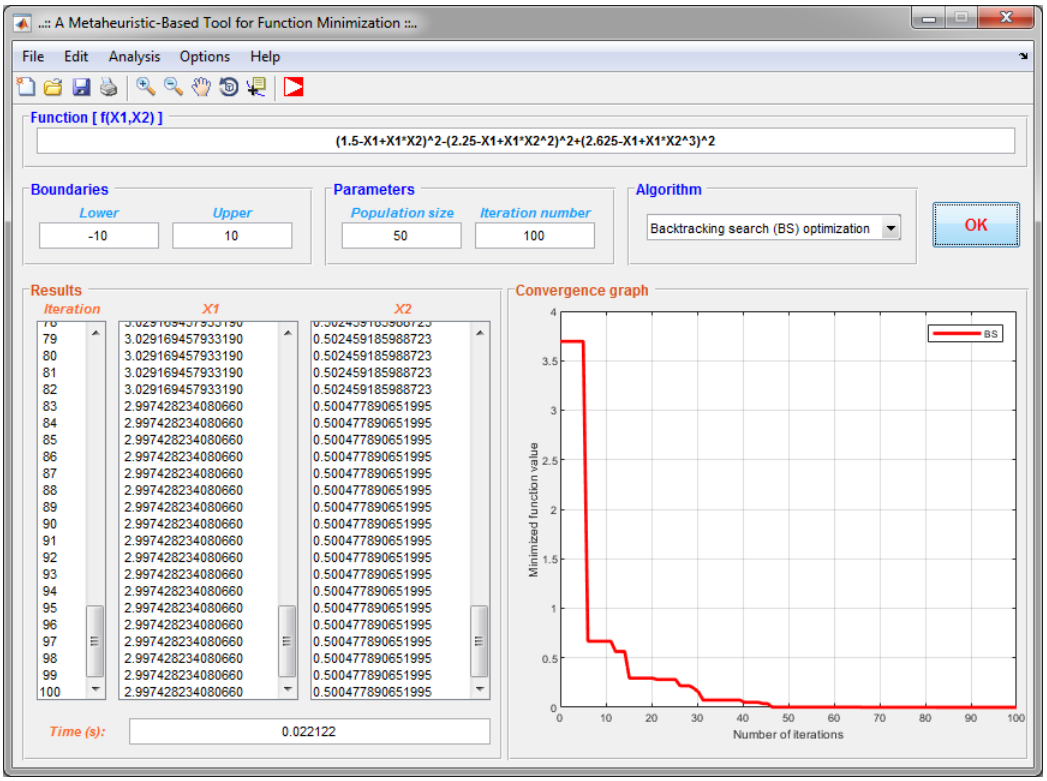

Figure 2. The screenshot of first application

In the second application, according to the results of Eq. 1, ABC and BS algorithms achieve similar value for X_1 and X_2 while HS finds different values for both variables. In terms of computational time, BS is in the first place with a run time of 0.026 sec., and VS is behind HS with a small difference of almost 0.01 sec. ABC is the slowest algorithm among all the algorithms, getting a run time of 0.367 sec. If the user considers the run-time, the user can choose BS since BS has similar X_1 and X_2 values with ABC, but it is almost 15 times quicker than ABC (Table 2). According to convergence graph given in Fig. 3, BS reaches its final solution after almost 15 iterations, which is slower than the rest of the algorithms.

Table 2. The comparative results for the second application

	ABC	BS	CS	FF	HS	VS
***X*1**	2.97628255	2.97983721	2.99961309	3.00004232	2.95510448	2.99999940
***X*2**	0.49397189	0.50036172	0.49999348	0.50000535	0.48802092	0.49999984
Time	0.36776180	0.02635010	0.24429006	0.87567287	0.22409545	0.03946317

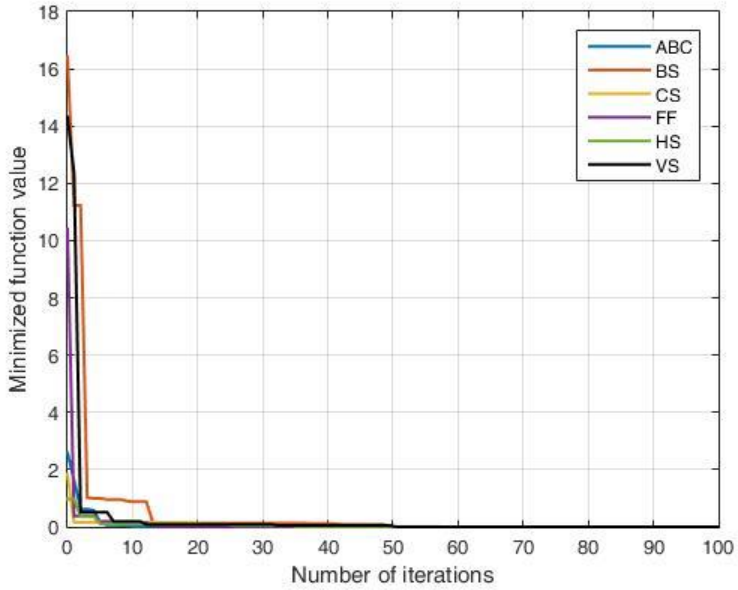

Figure 3. The convergence graph for the second application

In the third application, which can be shown in Table 3 according to the results of input equation [9]:

$$0.26 * (X_1^2 + X_2^2) - 0.48 X_1 X_2 \tag{2}$$

The algorithms achieve different results in terms of X_1 and X_2 values. However, considering run times, VS algorithm comes first and BS follows VS with a run time of 0.06 sec. CS and HS also have similar run time to get their final solutions, which are far from the run time obtained by VS and BS. The worst run time belongs to FF algorithm, which is almost 0.87 sec. BS and VS algorithms can be suitable more than the other opponents for this application (Table 3). The convergence graph given in Fig. 4, indicates that HS faces some difficulties in achieving its final solution while ABC shows its superiority by getting its final solution after almost 10 iterations.

Table 3. The comparative results for the third application

	ABC	BS	CS	FF	HS	VS
X1	-0.07916625	-0.02249523	3.47222e-04	-4.43756e-05	-0.02155195	-3.49869e-10
X2	-0.07660727	-0.02008269	2.79120e-04	-6.46705e-05	0.013621638	-5.77288e-10
Time	0.35227637	0.06424697	0.29807499	0.86906936	0.28008379	0.03804686

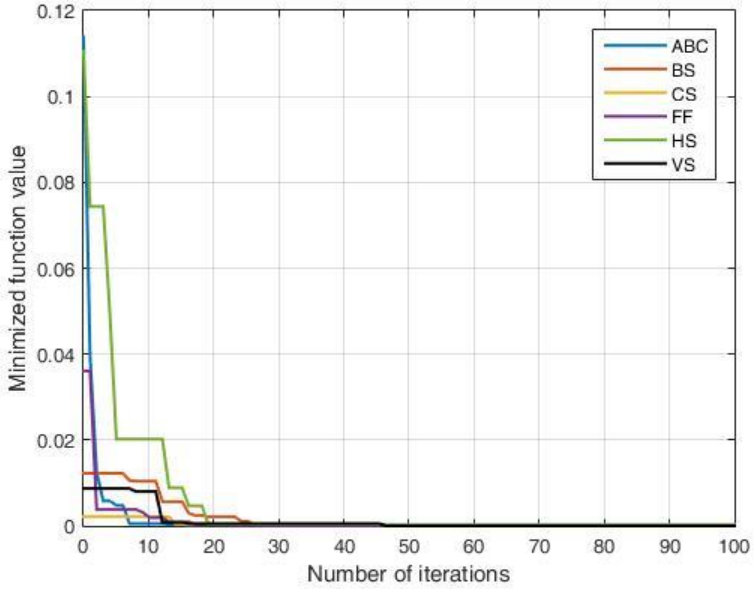

Figure 4. The convergence graph for the third application

In the fourth application, according to the results of input equation [9]:

$$4X_1^2 - 2.1X_1^4 + \frac{1}{3}X_1^6 + X_1X_2 - 4X_2^2 + 4X_2^4 \tag{3}$$

There are big differences in terms of the values of both X_1 and X_2 for all the algorithms. VS is able to find a solution faster than the other competitors, getting a run time of 0.03 sec, which is very similar run time obtained by BS. FF appears to be considerably slower than the other algorithms, having a run time of 0.91 sec. ABC algorithm also has the second-worst run time, and FF and ABC cannot be a good option for this application when regarding run times (Table 4). As can be seen from the convergence graph shown in Fig. 5, all the algorithms reach their final solutions without any big differences between them.

Table 4. The comparative results for the fourth application

	ABC	BS	CS	FF	HS	VS
X1	1.69120720	0.43466199	0.29621935	-0.00194452	0.36293924	-1.17934e-04
X2	-0.92438545	0.62900503	0.03489343	0.00171663	0.45480146	-1.33593e-04
Time	0.60208838	0.05967397	0.29621935	0.91013318	0.38014028	0.03416623

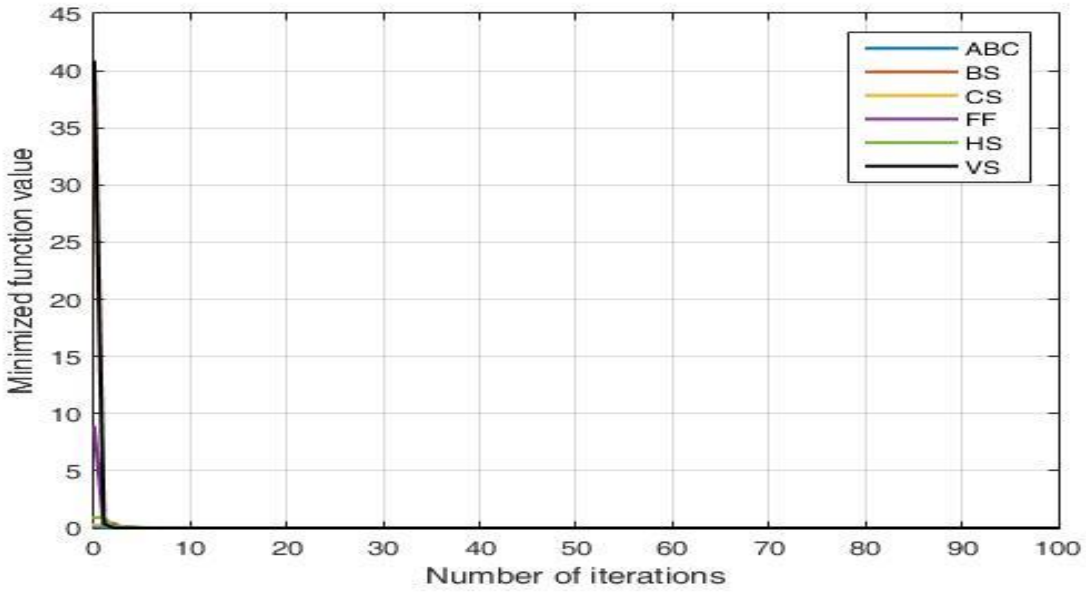

Figure 5. The convergence graph for the fourth application

4. Conclusions

Metaheuristic algorithms are used to solve many optimization problems due to its flexibility and ease of use. In this study, the tool is proposed for function minimization dependent on metaheuristic algorithms. The tool includes six metaheuristics, called ABC, BS, CS, FF, HS and VS, and is able to find optimal solution according to an input function and the parameters given by the user. The user can easily observe the results single or comparatively by using the user-friendly interface of the tool, and the convergence performance of the selected metaheuristic(s) is shown in the tool interface. To analyze the developed tool applicability and usability, four applications are chosen as examples. According to the results, the developed tool shows its advantage in achieving solutions for these examples fast and effectively. The tool can also help researchers to gain insight into analyzing the performance of the algorithms. In the future studies, on one hand, the scope of the tool should be extended by adding real-world problems. On the other hand, more effective metaheuristics can be integrated into the tool.

References

[1] Deb K. Multi-objective optimization using evolutionary algorithms. UK: Wiley; 2001.

[2] Haupt RL. Thinned arrays using genetic algorithms. IEEE Transactions Antennas Propagation 1994; 42:993-999.

[3] Rao RV, Pawar PJ. Modelling and optimization of process parameters of wire electrical discharge machining. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture 2009; 223:1431-1440.

[4] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. Journal of Global Optimization 2007; 39: 459-471.

[5] Civicioglu P. Backtracking search optimization algorithm for numerical optimization problems. Applied Mathematics and Computation 2013; 21:8121-8144.

[6] Yang XS, Deb S. Cuckoo search via levy flights. Nature & Biologically Inspired Computing 2009; 210:9-11.

[7] Yang XS. Firefly algorithm, levy flights and global optimization. Research and Development in Intelligent Systems 2010; 26:209-218.

[8] Yang XS. Harmony search as a metaheuristic algorithm. Music-inspired harmony search algorithm. Berlin: Springer; 2009.

[9] Doğan B, Ölmez T. A new metaheuristic for numerical function optimization: Vortex search algorithm. Information Science 2015; 293:125-145.

[10] MATLAB, The MathWorks, Inc. 2013.