Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

WEB BASED EDUCATIONAL TOOL FOR METAHEURISTIC ALGORITHMS

METASEZGİSEL ALGORİTMALAR İÇİN WEB TABANLI EĞİTİM ARACI

Güncel SARIMAN[1*], *Ecir Uğur KÜÇÜKSİLLE*[2]

[1]Electronic and Communication Engineering Department, Engineering Faculty, Süleyman Demirel University, 32100, Isparta.
guncelsariman@mu.edu.tr
[2]Computer Engineering, Engineering Faculty, Süleyman Demirel University, 32100, Isparta.
ecirkucuksille@sdu.edu.tr

Geliş Tarihi/Received: 30.05.2013, Kabul Tarihi/Accepted: 11.08.2013 doi: 10.5505/pajes.2014.15870
*Yazışılan yazar/Corresponding author

Özet

Günümüzde metasezgisel optimizasyon algoritmaları çok çeşitli optimizasyon problemlerinin çözümünde kullanılmaktadır. Bu algoritmalar matematiksel bir kanıta dayanmayıp, çoğunlukla doğada var olan olayların taklit edilmesiyle oluşturulmuş algoritmalardır. Bu çalışmada, web tabanlı eğitimsel metasezgisel test aracı geliştirilmiştir. Bu araç ile kullanıcılar, Yapay Bağışıklık ve Yapay Arı Kolonisi algoritmalarını Benchmark fonksiyonları üzerinde test edebilmekte, her bir algoritmanın parametrelerini değiştirerek optimizasyon sonuçlarını görebilmekte aynı zamanda kendi fonksiyonlarını da kısıtlarıyla beraber yazarak optimizasyon işlemini gerçekleştirebilmektedir. Ayrıca uygulamada her iki algoritmanın çalışma adımları hakkında da bilgi verilmektedir.

Anahtar kelimeler: Metasezgisel, Yapay Arı kolonisi, Yapay bağışıklık, Eğitim aracı.

Abstract

Metaheuristic optimization algorithms are nowadays being employed to solve a wide variety of optimization problems. These algorithms are not based on mathematical evidence and have mostly been developed by imitation of natural phenomenon. In this study, a web-based educational metaheuristics testing tool was developed. With this tool, the users are able to test Artificial Immune System and Artificial Bee Colony algorithms on Benchmark functions, observe the results of optimization by modifying the parameters for each algorithm, and at the same time perform optimization procedures by typing their own functions with their own constraints. In addition, information on the working steps of both algorithms are provided in the application.

Keywords: Metaheuristic, Artificial Bee colony, Artificial immune, Educational tool.

1 Introduction

Metaheuristic optimization algorithms are algorithms that have been developed to rapidly solve optimization problems with large spaces of candidate solutions. Just as there is no guarantee that these algorithms will provide a definite optimal solution, they can also provide different results in each problem. However, these algorithms are preferred by researchers as they provide solutions rapidly and require little mathematical knowledge for use. Genetic algorithms, tabu search algorithm, ant colony algorithm, artificial bee colony algorithm and artificial immune system algorithm are examples of existing metaheuristic algorithms, with new additions being developed with each passing day.

It cannot be expected for all existing metaheuristic algorithms to be effective in every optimization problem. In order to obtain the desired results, it is important to choose the algorithm that is suitable to the problem. When scanning the literature; it was observed that Lazaridis et al., [1] have developed in their studies web-based Linear Programming software for educational purposes. In the studies of Valdez et al., [2], it was observed that they have developed a toolbox which allowed for the algorithms called Genetic Algorithms, Particle Swarm Optimization and FPSO+FGA to be used in Matlab. However, web-based tool for the educational use of metaheuristic algorithms was not encountered in the literature. In the studies of Deng-xu and Rui-min, [20]. Cloud model-based Artificial Bee Colony Algorithm (CABC), developed through using positive cloud generator to generate the parameters that Artificial Bee Colony algorithm needed. The function test showed that CABC has better performance than the basic Artificial Bee Colony algorithm. CABC was used to solve the logistics location problem after discretization. Simulation results showed that it could solve this kind of problems better than the other algorithm. In the studies of Bi and Wang, [21]. They proposed a new ABC algorithm called Fast Mutation Artificial Bee Colony Algorithm, which modified selection scheme and proposed a mutation method instead of the behavior of scours. Simulation results on a set of benchmark functions have shown that the solutions produced by FMABC are better than those produced by the standard ABC algorithm.

Due to the ease of application of metaheuristic algorithms and the use of optimization in nearly all fields of science, education on metaheuristic algorithms are nowadays being provided as courses in many departments, or as a subject in various courses. With only the working steps of these algorithms being generally described to the students, the students are not provided with the opportunity to practice these algorithms. The application of heuristic model of teaching in Tehnical and IT Education allows to achieve greater effects of teaching and learning, and their application in practice in education for action in emergency situations [3].

In this study, a web-based educational software was developed in which only the Artificial Immune System, [4] and Artificial Bee Colony [5], [6], [7] can be optimized at the present, in which changes in the results can be observed by modifying the algorithm parameters, and in which the students can optimize by typing their own functions. In addition, textual and visual materials on the working steps for these algorithms were provided in the software that was developed.

The article was presented as follows: In the first section, general information regarding the study was provided. Test function formulization were provided in the second section, artificial bee colony algorithm in the third section, artificial immune system algorithm in the fourth section, web-based education tool in the fifth section, optimization of algorithms in the sixth section, results in the seventh section, and finally the references were provided.

2 The Formulation of Optimization Functions

To find the optimal solution for a problem, it is a common practice to use evolutionary calculation methods inspired from nature, and to optimize the problem with different algorithms in this field. Benchmark functions are employed as proof of validity for an algorithm in this field, and also to measure the algorithm by applying it to any problem. Test functions used in optimization algorithms are divided into 4 classes. These classes are as follows: Unimodal, convex and multidimensional; multimodal, two-dimensional with a small number of local extremes; multimodal, two-dimensional with large number of local extremes; and finally multimodal, multidimensional with large number of local extremes [8]. In this study, the Rastrigin, Six-hump Camel Back, Schwefel, Schubert, Drop Wave functions, frequently used in the literature, can be employed in optimization with the educational tool.

The Rastrigin function is used to determine local minima. For this reason, this function belongs to the multimodal class. The mathematical definition of the function is as shown in equation (1).

$$f(X) = 10 * n + \sum_{i=1}^{n} \left[x_i^2 - 10 * cos(2 * \pi * x_i)\right] \qquad (1)$$

The variables for the Rastrigin function are between -5.12≤x_i ≤5.12, and i=1,…,n represent the size value. The global minimum value of the function is f(x)=0 and the variables are x_i=0. In Figure 1 a two-dimensional representation of the Rastrigin function is provided.

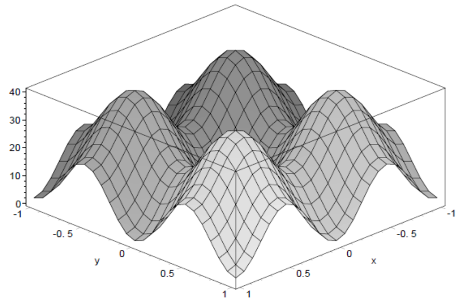


Figure 1: Rastrigin function.

Schwefel's function is a function which has the second best minimum distant to the global minimum [9]. They are inclined to deviate in the wrong direction in search algorithms. The mathematical definition for the Schwefel's function is provided in equation 2.

$$\mathrm{f(x)} = \sum_{i=1}^{n} \left[-x_i * \sin\left(\sqrt{|x_i|}\right)\right] \qquad (2)$$

Variables for the Schwefel's function are between -500≤x_i ≤500 and i=1,…, represents the n size value. The global minimum value of the function is $\mathrm{f(x)} = -418.9829n$ and the variables are x_i=420.9687. In Figure 2, a two-dimensional representation of the Schwefel's function is provided.

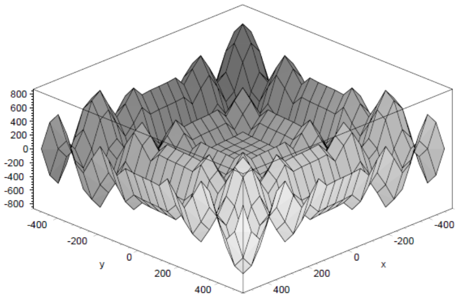


Figure 2: Schwefel's function.

The Schubert function is a multimodal test function. A representation of this function with two variables in provided in equation 3.

$$\mathrm{f}(X_1, X_2) = -\sum_{i=1}^{5} i * \cos\big((i+1) * x_1 + 1\big) * \sum_{i=1}^{5} i * \cos((i+1) * x_2 + 1) \qquad (3)$$

Variable ranges are limited between -5.12≤x_1≤5.12 and -5.12≤x_2≤5.12. In Figure 3, a two-dimensional representation of the function is provided.

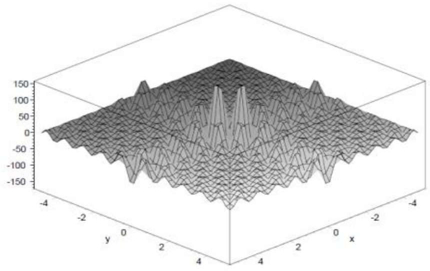


Figure 3: Schubert function.

The Drop Wave function belongs to the multimodal test function class. This function only has two variables. Its mathematical expression is provided in equation 4.

$$\mathrm{f}(X_1, X_2) = -\frac{1 + \cos(12 * \sqrt{x^2 + x_2^2})}{\frac{1}{2}(x_1^2 + x_2^2) + 2} \qquad (4)$$

Variable ranges are limited between -5.12≤ x_1≤5.12 and -5.12≤ x_2≤5.12. In Figure 4, a two-dimensional representation of the function is provided.

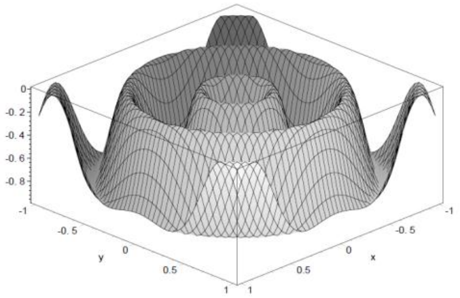


Figure 4: Drop Wave function.

The Rastrigin and Six-hump Camel Back functions are difficult problems with many local minima compared to the other functions. The Six-Hump Camel Back function was used to compare the performance of the artificial bee colony and artificial immune system algorithms in the optimization section of the article. The Six-hump Camel Back function is a global optimization function. The function has 6 local minimum, of which 2 possess global minimum values. This function with two variables is as shown in equation (5).

$$\mathrm{f}(X_1, X_2) = \left(4 - 2.1 * X_1^2 + \frac{X_1^4}{3}\right) * X_1^2 + X_1 * X_2 + (-4 + 4 * X_2^2) * X_2^2 \qquad (5)$$

In optimization procedures, the lower and upper limits for equation 5 in the literature [10] are limited between -3≤ X_1 ≤3 and -2 ≤ X_2 ≤2. The function value for the Global minimum problems is $\mathrm{f}(X_1) = -1{,}0316$, and for the variable global minimum values it is $(X_1, X_2) = (-0.0898, 0.7126)$. Figure 5 provides a representation for the Six-hump Camel Back function.

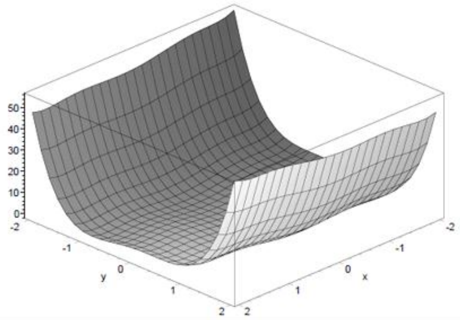


Figure 5: Six-Hump camel back function.

3 Artificial Bee Colony Algorithm

In the optimization of non-linear problems, it is not possible to reach a solution by using simple mathematical expressions. It has been proven by scientists that benefiting from swarm behavior of living beings in nature instead of solutions that require intensive mathematical operations with derivative expressions can lead to solutions with relatively shorter programming. The artificial bee colony algorithm is an optimization algorithm based on swarm intelligence and inspired from the food searching behavior of honey bee colonies [5]. Bees live in colonies and display behavior that can be taken as example for problem solving. Honey bee colonies can distribute tasks among themselves and provide adaptive responses to environmental changes [11].

In the algorithm there are fundamentally 3 types of bees, which are the worker bees, the onlooker bees and the explorer bees. The distribution of tasks is achieved with the workers bees gathering food sources, with the onlooker bees following the dance of worker bees to select food sources, and with the explorer bees performing random searches. The onlooker and worker bees are distributed equally, half and half, within the population. In the overall function logic of the algorithm, the worker bees first calculate nectar quantities with the aid of the fitness function by traveling to the food sources. In ABC algorithm, each food source represents a probable solution in proportion to the nectar quantity [12]. Afterwards, they work to find new and preferable food sources within the neighborhood of their food sources. Scout bees are finally sent to randomly discover new food sources. By hiding the best solution, this cycle continues until the termination conditions are met. The most important difference of the explorer bees from the others is that, while worker and onlooker bees perform local searches the explorer bees perform global searches. The parameters and tools of use that need to be determined prior to the algorithm use are as follows: the iteration number to determine the number of runs for the algorithm to be optimized, the limit parameter for the transformation of a worker bee displaying no progress in the iterations into an explorer bee, and finally the number of worker bees to locate the food source must be determined. In the first stage of the algorithm, random values between the lower and upper values of the function parameters to be optimized will be determined with the aid of equation (6) in order to determine (the food sources of) the worker bees:

$$X_{ij} = X_j^{min} + rand(0{,}1) * (X_j^{max} - X_j^{min}) \qquad (6)$$

I index in equation (6) represents the food source number, j index represents the parameter to be optimized. In the beginning stage, bees determine the quantity of nectar by finding the fitness values. The worker bees are tasked with finding new food sources in their neighborhood. In case the lower and upper values of the parameters are exceeded, minimum and maximum values will be used. The formula for finding new food sources is shown in Equation 7.

$$V_{ij} = X_{ij} + \varphi_{ij} * (X_{ij} - X_{kj}) \qquad (7)$$

In Equation 7, φ_{ij} is the random number maintained between -1 and 1, while X_{kj} is the randomly chosen parameter value from the sources belonging to the j parameter. Following this, the fitness values of new sources are determined using equation 8. Worker bees determine the sources they will travel to by using these fitness values in the roulette wheel. Also, in every iteration the non-development situations are determined by comparing the bees with the worker bees.

$$Uym(x) = \begin{cases} \dfrac{1}{1 + f(X)}, & f(X) \geq 0 \\ 1 + |f(X)|, & f(X) < 0 \end{cases} \qquad (8)$$

The chance of selection of each worker bee is determined by dividing the worker bees with known fitness values with the sum of all of the fitness values. By using equation 7, the processes continue until all of the new onlooker bees select a food source. In the scout bee stage, by controlling the non-development situation of the bees, a new food source is discovered in all space for those exceeding the limit value.

4 Artificial Immune System Algorithm

Artificial immune systems are general purpose intuitive procedures developed by imitating nature, as is the case with genetic algorithms and swarm intelligence [7]. The artificial immune system is based on the working principle of the natural immune system in the human body. The human immune system is an amazing natural defense mechanism that protects and teaches the body against foreign substances [12]. The system functions on the principle of identifying and neutralizing foreign cells that enter the body with lymphocytes. Lymphocytes have B and T cells, and the B tells are tasked with the production of antibodies. When a foreign molecule enters the body, special molecular (macrophages) surround this foreign molecule and enable other immune cells to see them. When helper "T" cells see these special molecules, they stimulate white blood cell production and these cells begin to proliferate. "B" cells produce chemical substances called antibodies, and the antibodies facilitate the destruction of the antigen by combining with it. Killer "T" cells stimulated by the helper "T" cells ensure the destruction of the antigen. The Immune response is displayed in this manner [14]. The artificial immune system algorithm was developed by modelling the defense mechanism in living beings, and has served in various fields as a guide in problem solving. Owing to its use of double coding, to the presence of the fitness function and to being subject to mutation, the artificial immune system algorithm shows similarity to genetic algorithms. Artificial immune system has provided better results than genetic algorithm and other optimization techniques in fields such as pattern recognition, computer security, network security and optimization, etc. [15], [16]. The procedure generally employed in optimization problems is the clonal selection algorithm. Only cells recognizing the antigen proliferate, to the extent they can recognize the antigen. In other words, recognizing cells are selected over non-recognizing cells [17]. With this procedure, the aim is to approach the desired situation with each iteration. The stages of clonal selection algorithm are as follows:

Step 1: Form a random starting population consisting of antibodies.

Step 2: According to the fitness function, calculate the fitness values and stimulation level of the antibodies.

Step 3: Select n number of antibodies that possess the highest similarity.

Step 4: By considering the degree of similarity, perform cloning procedure with the tournament selection procedure.

Step 5: Mutate antibodies with high degree of similarity to reduce the degree of similarity.

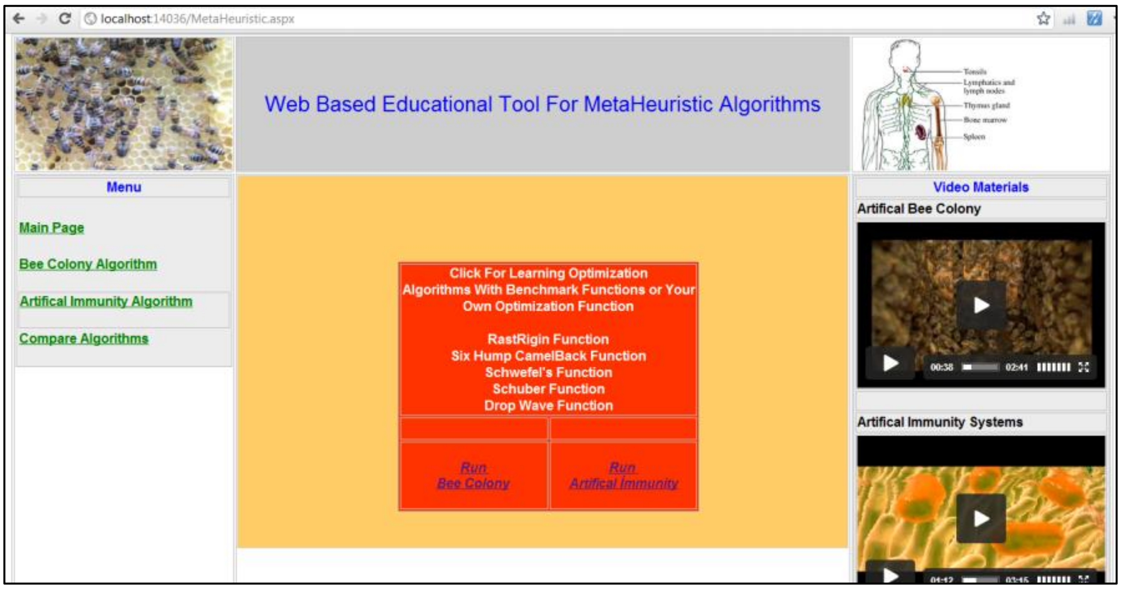
Step 6: Also calculate the degree of similarity of antibodies in the sub-population. Substitute the antibodies with the highest degree of similarity with n antibodies showing the least similarity [18]. The parameters and aims that need to be determined prior to the algorithm are as follows: function minimum and maximum optimization problem solving type optimization type with, number of runs for the algorithm to be optimized with iteration number and, antibody number with antibody parameter determined.

5 Web Based Educational Tool

Web based education is generally considered as an e-learning concept. With the increasing prevalence of web sites nowadays, the internet plays an active role at every stage of education. At the same time designing, developing, applying and evaluating learning systems that are open, flexible and decentralized, and publishing course content online according to the various sizes of online learning environments have allowed web-based education programs to be commonly used. In this study, a new web interface was developed to facilitate the teaching of metaheuristic algorithms to students. By means of the simulation, the student can put into practice the theoretical concepts studied during the course [19].

Web based educational tool was developed in. Net Framework 4.0 that based in Microsoft architecture. In web platform Asp.Net technology and JavaScript was used. For statistics, all test and applications are recorded to the database. So that Microsoft Sql Server 2012 was used in database management.

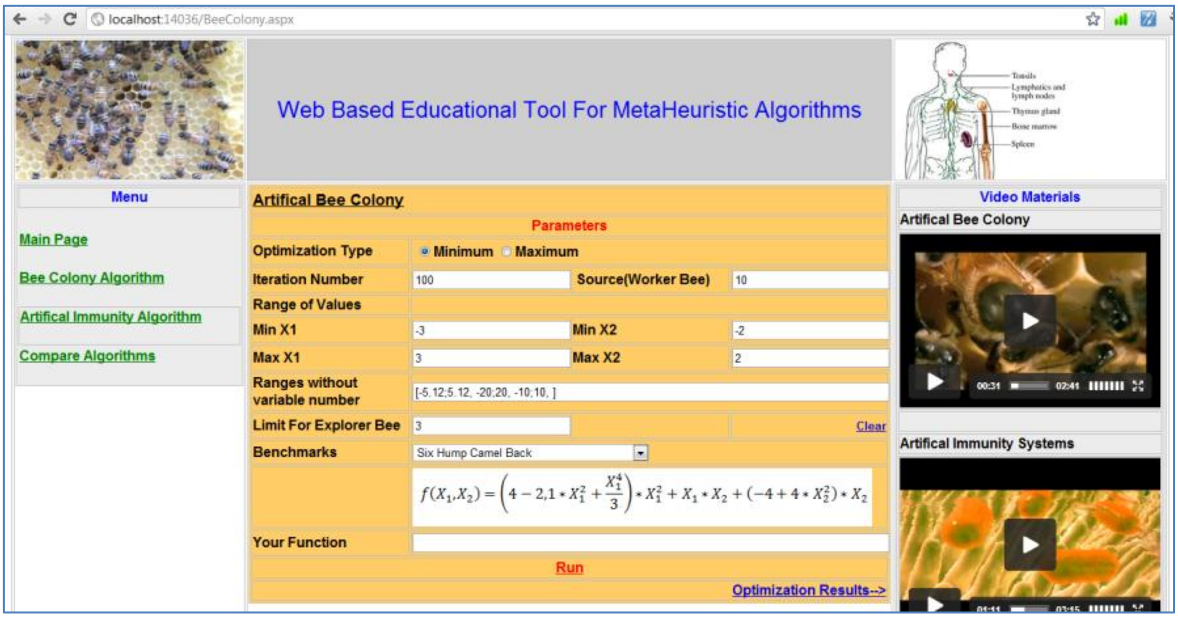
With the education software that was developed, the students are able to optimize the five benchmark test functions or their own functions by choosing the artificial bee colony or artificial immune system algorithms, from any place with internet access and by performing entries at any time. In addition, by supporting the education on algorithms with video displays, the web based educational tool facilitates training for its users. The lecturer or teacher can illustrate with examples the concepts of evolutionary computation or bio-inspired optimization by using the toolbox in the graduate or undergraduate course [2]. The user page which opens the entry screen can be easily accessed from the menu on the left side, while the video materials can be readily accessed from the right side. At the same time, optimization can be performed by choosing the "Run Artificial Immunity" or "Run Bee Colony" links on the entry screen, and two algorithms can be compared by using the "Compare Algorithms" button.



Click For Learning Optimization Algorithms With Benchmark Functions or Your Own Optimization Function

RastRigin Function
Six Hump CamelBack Function
Schwefel's Function
Schuber Function
Drop Wave Function

Figure 6: Overview of the education set.



Web Based Educational Tool For MetaHeuristic Algorithms

Artifical Bee Colony

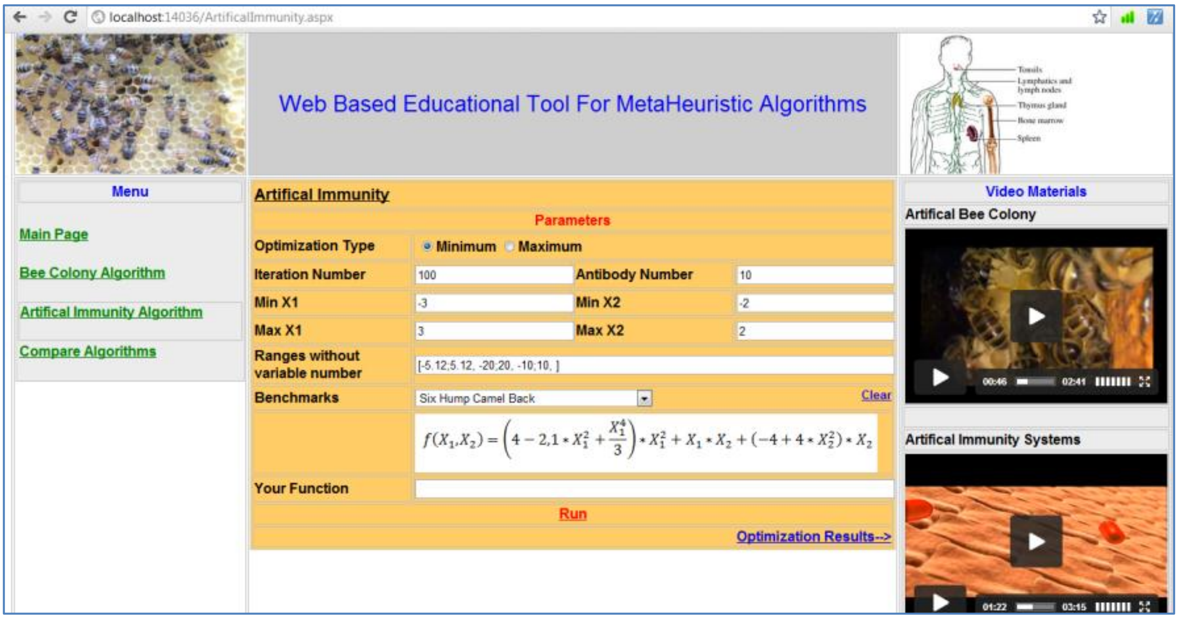
Parameters

Optimization Type	● Minimum ○ Maximum		
Iteration Number	100	Source(Worker Bee)	10
Range of Values			
Min X1	-3	Min X2	-2
Max X1	3	Max X2	2
Ranges without variable number	[-5.12;5.12, -20;20, -10;10,]		
Limit For Explorer Bee	3		Clear
Benchmarks	Six Hump Camel Back		
	$f(X_1,X_2)=\left(4-2{,}1*X_1^2+\frac{X_1^4}{3}\right)*X_1^2+X_1*X_2+(-4+4*X_2^2)*X_2$		
Your Function			

Run

Optimization Results-->

Figure 7: Artificial bee colony optimization page.



Web Based Educational Tool For MetaHeuristic Algorithms

Artifical Immunity

Parameters

Optimization Type	● Minimum ○ Maximum		
Iteration Number	100	Antibody Number	10
Min X1	-3	Min X2	-2
Max X1	3	Max X2	2
Ranges without variable number	[-5.12;5.12, -20;20, -10;10,]		
Benchmarks	Six Hump Camel Back		Clear
	$f(X_1,X_2)=\left(4-2{,}1*X_1^2+\frac{X_1^4}{3}\right)*X_1^2+X_1*X_2+(-4+4*X_2^2)*X_2$		
Your Function			

Run

Optimization Results-->

Figure 8: Artificial immune system algorithm optimization page.

In Figure 7, the parameters of the artificial bee colony can be seen. Many parameters such as the iteration number, the number of worker bees can be controlled by the user.

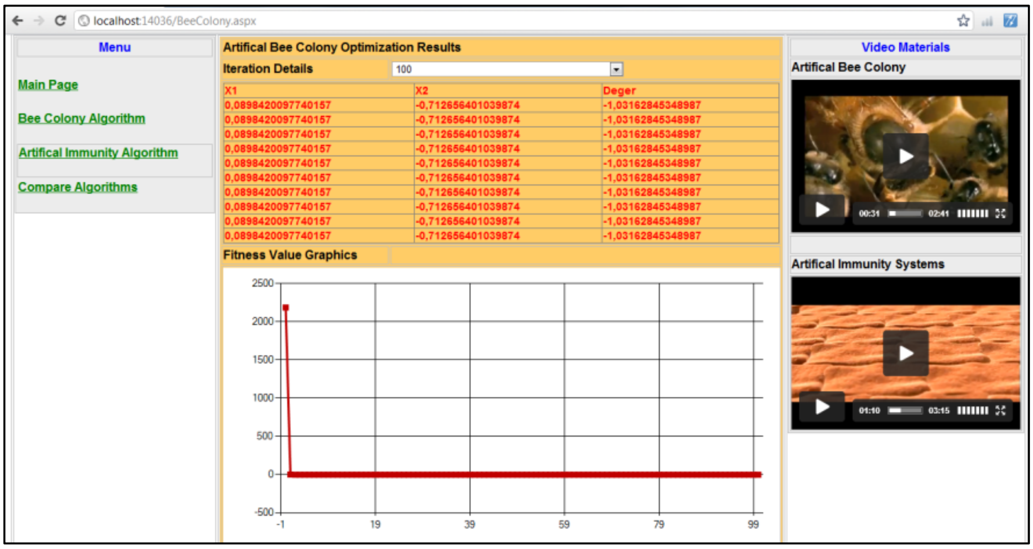
In addition, the minimum and maximum values of the functions to be tested appear as default 2 parameters for the users. If further variables are to be optimized, min and max values can be entered within the brackets in the "Range without variable number" section, by typing a semi colon in between the values. The number of scout bees is set using the "limit for explorer bee" section. In the final section, it is possible to perform the optimization of the previously provided five test functions, or of the function that can be typed by the user. With "optimization type," it is possible to choose whether maximization or minimization optimization will be performed. The test functions are Rastrigin, Six-hump Camel Back, Drop Wave, Schubert's and Schwefel's, respectively.

Optimization was performed with the educational tool using the artificial immune system algorithm and Six-hump Camel Function. The optimization screen appears to the users by entering the "Run immunity" section on the entry screen. On Figure 8, the interface and parameter entries for the artificial immune system algorithms can be seen.

Similar to the artificial bee colony on the user interface, the parameters can be controlled by user. The iteration number by using "iteration number," the antibody number by using "antibody", which one of the maximization or minimization optimization types will be used by using the "Optimization type" button, and finally the minimum and maximum values of the variables can all be determined by the user. To optimize more variables, min and max values can be entered within the brackets into the "Range without variable number" section, by typing a semi colon between the values. The benchmark test function that can be chosen by the user or function fields that can be typed by the user are indicated on the interface. Once the Run operation is performed, the next stage commences.

6 Optimization Results of Algorithm

After the “Run” operation is performed for the artificial bee colony, the test results can be visualized in the next stage (“Optimization Results”). In Figure 9, the results of the performed test can be seen by choosing the Six-hump Camel Back function. For the two variables, the value range was determined as -5,12 and 5,12. The function and parameters used for the bee colony algorithm can be seen on Table1.



Artifical Bee Colony Optimization Results

Iteration Details: 100

X1	X2	Deger
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987
0,0898420097740157	-0,712656401039874	-1,03162845348987

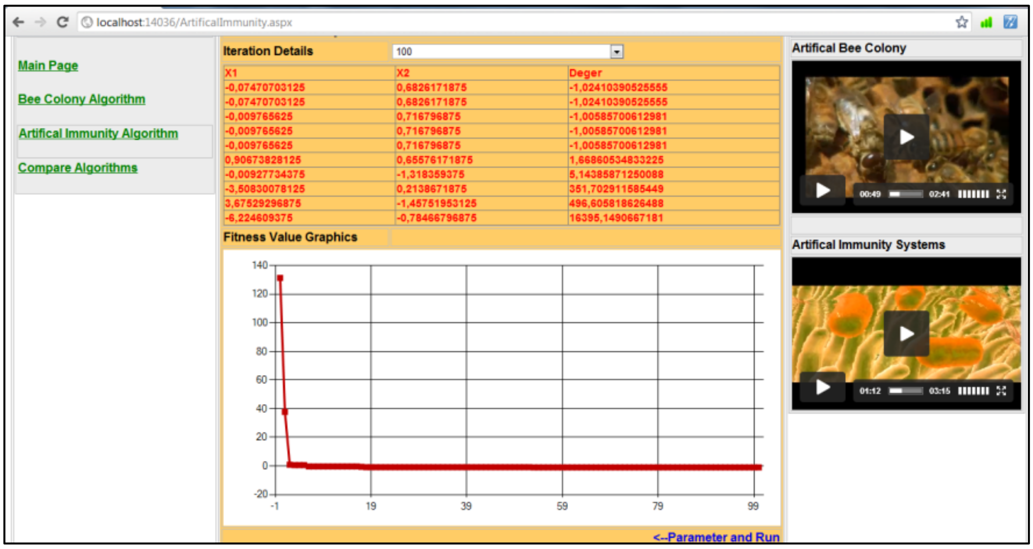
Figure 9: Artificial bee colony optimization results.

Table 1: Most suitable parameters used for the artificial bee colony algorithm.

Iteration Number	100	Food Source Number	10
Min X1	-3	Max X1	3
Min X2	-2	Max X2	2
Explorer Bee Number	3		
Function		$\mathrm{f}(X_1, X_2) = \left(4 - 2.1 * X_1^2 + \frac{X_1^4}{3}\right) * X_1^2 + X_1 * X_2 + (-4 + 4 * X_2^2) * X_2^2$	

Table 2: Most suitable parameters used for the artificial immune system algorithm.

Iteration Number	1000	Antibody Number	10
Min X1	-3	Max X1	3
Min X2	-2	Max X2	2
Function		$\mathrm{f}(X_1, X_2) = \left(4 - 2.1 * X_1^2 + \frac{X_1^4}{3}\right) * X_1^2 + X_1 * X_2 + (-4 + 4 * X_2^2) * X_2^2$	



Iteration Details: 100

X1	X2	Deger
-0,07470703125	0,6826171875	-1,02410390525555
-0,07470703125	0,6826171875	-1,02410390525555
-0,009765625	0,716796875	-1,00585700612981
-0,009765625	0,716796875	-1,00585700612981
-0,009765625	0,716796875	-1,00585700612981
0,90673828125	0,65576171875	1,66860534833225
-0,00927734375	-1,318359375	5,14385871250088
-3,50830078125	0,2138671875	351,702911585449
3,67529296875	-1,45751953125	496,605818626488
-6,224609375	-0,78466796875	16395,1490667181

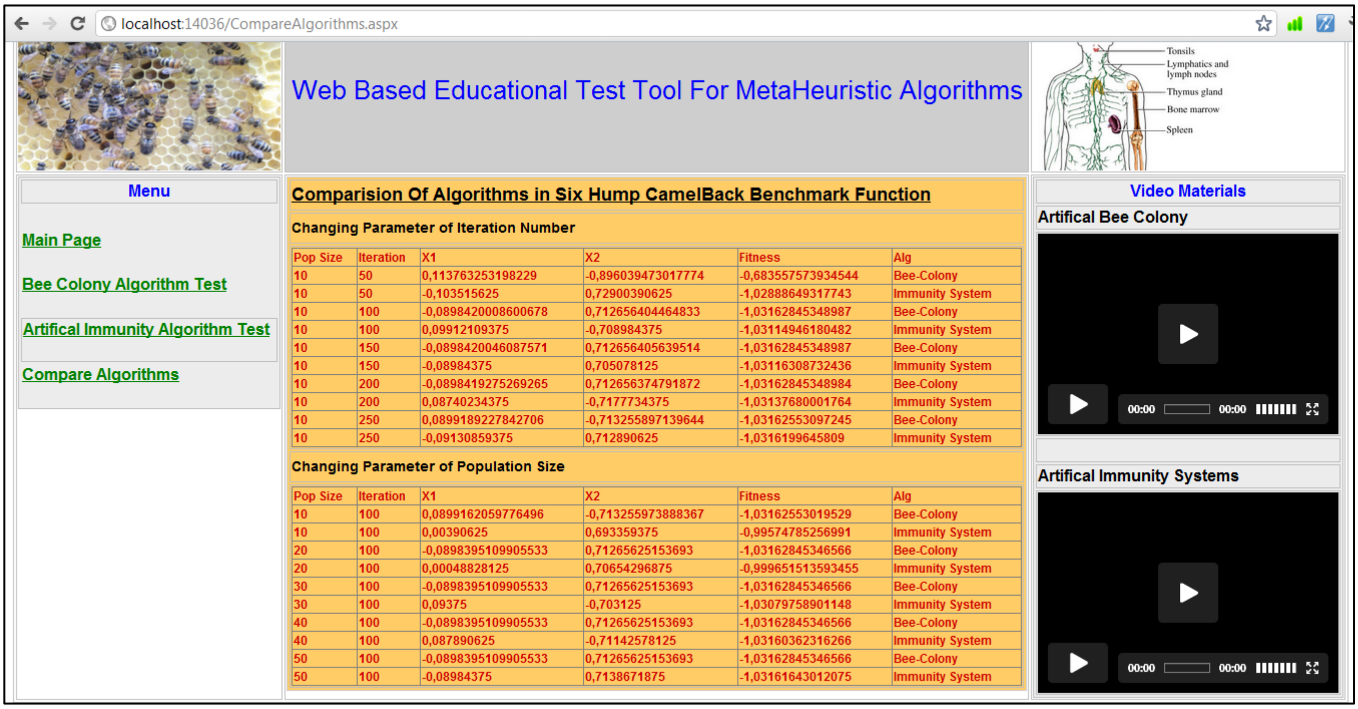
Figure 10: Artificial immune system optimization results.

On Figure 9, the value of each worker bee in the iterations is listed below the “iteration details” header. The best fitness values obtained during the iterations with the fitness value graphics are shown as a graph. The results obtained after the test for artificial immune system algorithm was performed are shown on Figure 10. The used parameters are displayed on Table 2.

On Figure 10, the fitness value of each worker bee in the iterations is listed below the “iteration details” header. The best fitness value obtained during the iterations with the fitness value graphics are shown as a graph. The test results obtained from the study can be seen in the “Compare algorithms” section. The fitness values in the comparison have been indicated on the table. First, the iteration number was gradually increased while maintaining the population number constant. In the second stage, the population number was gradually increased while maintaining the iteration number constant. Figure 11 displays the comparison window.

When reviewed in detail, the test performed using the Six-Hump Camel Back function is provided on table 3 with a change in the iteration number. On Table 3 it is seen that, by changing the iteration number to 50, 100, 150, 200, 250, the best result is approached as the value is increased. It was observed that the artificial bee colony algorithm provides better results at 100 iteration and 10 population size. With the same function and population size, the artificial immune system algorithm obtained the best results at 250 iteration. In brief, shorter periods meant better results.

Table 4 shows the results obtained by keeping the iteration number constant. Tests performed using the Six-Hump Camel Back function was provided in terms of changes in population numbers. With Table 4 it is seen that, by changing the population number to 10, 20, 30, 40, 50, the best result is approached as increased. It can be seen on Table 4 that the artificial bee colony algorithm provides the best result in literature with a population size of 10, while on the other the artificial immune system approaches the best result when the population number is 50.


localhost:14036/CompareAlgorithms.aspx
Tonsils
Lymphatics and lymph nodes
Thymus gland
Bone marrow
Spleen
Menu
Main Page
Bee Colony Algorithm Test
Artifical Immunity Algorithm Test
Compare Algorithms
Video Materials
Artifical Bee Colony
00:00
00:00
Artifical Immunity Systems
00:00
00:00

Web Based Educational Test Tool For MetaHeuristic Algorithms

Comparision Of Algorithms in Six Hump CamelBack Benchmark Function

Changing Parameter of Iteration Number

Pop Size	Iteration	X1	X2	Fitness	Alg
10	50	0,113763253198229	-0,896039473017774	-0,683557573934544	Bee-Colony
10	50	-0,103515625	0,72900390625	-1,02888649317743	Immunity System
10	100	-0,0898420008600678	0,712656404464833	-1,03162845348987	Bee-Colony
10	100	0,09912109375	-0,708984375	-1,03114946180482	Immunity System
10	150	-0,0898420046087571	0,712656405639514	-1,03162845348987	Bee-Colony
10	150	-0,08984375	0,705078125	-1,03116308732436	Immunity System
10	200	-0,0898419275269265	0,712656374791872	-1,03162845348984	Bee-Colony
10	200	0,08740234375	-0,7177734375	-1,03137680001764	Immunity System
10	250	0,0899189227842706	-0,713255897139644	-1,03162553097245	Bee-Colony
10	250	-0,09130859375	0,712890625	-1,0316199645809	Immunity System

Changing Parameter of Population Size

Pop Size	Iteration	X1	X2	Fitness	Alg
10	100	0,0899162059776496	-0,713255973888367	-1,03162553019529	Bee-Colony
10	100	0,00390625	0,693359375	-0,99574785256991	Immunity System
20	100	-0,0898395109905533	0,71265625153693	-1,03162845346566	Bee-Colony
20	100	0,00048828125	0,70654296875	-0,99965151359345 5	Immunity System
30	100	-0,0898395109905533	0,71265625153693	-1,03162845346566	Bee-Colony
30	100	0,09375	-0,703125	-1,03079758901148	Immunity System
40	100	-0,0898395109905533	0,71265625153693	-1,03162845346566	Bee-Colony
40	100	0,087890625	-0,71142578125	-1,03160362316266	Immunity System
50	100	-0,0898395109905533	0,71265625153693	-1,03162845346566	Bee-Colony
50	100	-0,08984375	0,7138671875	-1,03161643012075	Immunity System

Figure 11: Comparison interface.

Table 3: Results obtained by maintaining the population number constant.

Pop Size	Iteration	X1	X2	Fitness	Algorithm
10	50	0,113763253	-0,896039473	-0,683557573	Bee-Colony
10	50	-0,10351562	0,7290039062	-1,028886493	Immunity System
10	100	-0,08984200	0,7126564044	-1,031628453	Bee-Colony
10	100	0,099121093	-0,708984375	-1,031149461	Immunity System
10	150	-0,08984200	0,7126564056	-1,0316284534	Bee-Colony
10	150	-0,08984375	0,705078125	-1,0311630873	Immunity System
10	200	-0,08984192	0,71265637479	-1,0316284534	Bee-Colony
10	200	0,087402343	-0,7177734375	-1,0313768000	Immunity System
10	250	0,089918922	-0,7132558971	-1,03162553097	Bee-Colony
10	250	-0,09130859	0,712890625	-1,03161996458	Immunity System

Table 4: Results obtained by maintaining the iteration number constant.

Pop Size	Iteration	X1	X2	Fitness	Algorithm
10	100	0,089916205	-0,713255973	-1,0316255301	Bee-Colony
10	100	0,00390625	0,693359375	-0,9957478525	Immunity System
20	100	-0,08983951	0,7126562515	-1,0316284534	Bee-Colony
20	100	0,000488281	0,7065429687	-0,9996515135	Immunity System
30	100	-0,089839510	0,7126562515	-1,0316284534	Bee-Colony
30	100	0,09375	-0,703125	-1,0307975890	Immunity System
40	100	-0,089839510	0,7126562515	-1,0316284534	Bee-Colony
40	100	0,087890625	-0,711425781	-1,0316036231	Immunity System
50	100	-0,089839510	0,7126562515	-1,0316284534	Bee-Colony
50	100	-0,08984375	0,7138671875	-1,0316164301	Immunity System

7 Conclusions

In this study Web-based metaheuristic optimization software was developed. Thanks to its flexible interface, the application possesses an interface in which various parameters can be modified. A function with N variable can be identified and the value range of each variable and be entered. In addition, the effects of the parameters on optimization can be observed by modifying the parameters for each optimization. It is possible to compare the results of the algorithms by optimizing the same function with different optimization algorithms. As the developed application is an internet program, it enables the students to access the system at any hour, and allows for an entertaining and easier learning experience with the video materials it provides.

Developed educational tool, used in Heuristic algorithm, graduate lesson of Computer Engineering department. In line with the feedback from the students about educational tool, this has been shown to contribute to the expression of graduate lesson and was decided to use by lesson's teacher.

In further stages of the study, the adaptation of different metaheuristic algorithms to the system will be ensured.

8 References

[1] Paparrizos, V. K., Samaras, N. and Sifaleras, A., "Visual LinProg: A Web-based Educational Software for Linear Programming", Computer Application in Engineering Education, 15 (1), pp. 1-14, 2007.

[2] Valdez, F., Melin, P. and Castillo, O., "Toolbox for Bio-Inspired Optimization of Mathematical Functions", Computer Application In Engineering Education, 2011.

[3] Beres, K., "Distance learning, heuristic model of education and alternative energy sources with liquid battery", Technics Technologies Education Management, 7 (3), pp. 1418-1426, 2012.

[4] De Castro, L. N. and Zuben, F. J. V., "Artificial Immune Systems: Part-II A Survey of Applications", Technical Report, 2000.

[5] Karaboğa, D., "An idea based on honey bee swarm for numerical optimization", Technical Report TR06, Erciyes University, Turkey, pp. 1-6, 2005.

[6] Linh, N. T. and Anh, N. Q., "Application artificial bee colony algorithm (ABC) for reconfiguring distribution network", Second International Conference on Computer Modeling and Simulation, 1, pp. 102-106, 2010.

[7] Kang, F., Li, J. and Ma, Z., "Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions", Information Sciences, 181 (16), pp. 3508-3531, 2011.

[8] Molga, M. and Smutnicki, C., "Test functions for optimization needs", http://www.zsd.ict.pwr.wroc.pl/files/docs/functions.pdf, 2005.

[9] Ökdem, S., Karaboğa, D., "Gerçek Zamanlı Optimizasyon İçin Gelişime Dayalı Hızlı Bir Algoritma", 2005.

[10] Broeck, G. V. D. and Driessens, K., "Automatic Discretization of Actions and States in Monte-Carlo Tree Search", International Workshop on Machine Learning and Data Mining in and around Games (DMLG). 2nd Ed., Athens, pp. 1-12, 2011.

[11] Karaboğa, D. and Akay, B., "A Survey: Algorithms Simulating Bee Swarm Intelligence", Springer, 31 (1-4), pp. 61-85, 2010.

[12] Yang, L., Boxue, T. and Xue, Z., "Position Accuracy Improvement of PMLSM System Based on Artificial Immune Algorithm", Proceedings of the 26th Chinese Control Conference, Zhangjiajie, Hunan, China, pp. 3679-3683, 2007.

[13] Gao, W., Liu, S. and Huang, L., "Global best artificial bee colony algorithm for global optimization", Journal of Computational and Applied Mathematics, 236 (11), pp. 2741-2753, 2012.

[14] Engin, O. and Döyen, A., "Artifical Immune Systems And Applıcatıons In Industrial Problems", G.U. Journal of Science, 17 (1), pp. 71-84, 2004.

[15] De Castro, L. N. and Timmis, J., "Artificial Immune Systems: A novel paradigm to pattern recognition", Artificial Neural Networks in Patttern Recognition, 2, pp. 67-84, 2002.

[16] De Castro, L. N., and Von Zuben, F. J., "The Clonal Selection Algorithm with Engineering Applications", Workshop on Artificial Immune Systems and Their Applications. Las Vegas, USA, pp. 36-37, 2000.

[17] Karaboğa, D., "Yapay Zeka Optimizasyon Algoritmaları", Istanbul, Atlas Press, 2004.

[18] Castro, L. N. and Zuben, J. V., "Learning and Optimization Using the Clonal Selection Principle", IEEE Transactions on Evolutionary Computation, 6 (3), pp. 239-251, 2002.

[19] Mendez, J. A., Lorenzo, C., Acosta, L., Torres, S. and Gonzales, E., "A Web-Based Tool for Control Engineering Teaching", Computer Application In Engineering Education, 14 (3), pp. 178-187, 2006.

[20] Deng-xu, H., Rui-min, J., "Cloud model-based Artificial Bee Colony Algorithm's Application in The Logistics Location Problem", Information Management, Innovation Management and Industrial Engineering (ICIII), 2012 International Conference on, pp. 256-259, 2012.

[21] Bi, X., Wang, Y., "An Improved Artificial Bee Colony Algorithm", Computer Research and Development (ICCRD), 2011 3rd International Conference on, pp. 174-177, 2011.