International Journal of Innovative Computing, Information and Control
Volume **8**, Number **9**, September **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 6107–6121

A NOVEL ARTIFICIAL BEE COLONY-BASED ALGORITHM FOR SOLVING THE NUMERICAL OPTIMIZATION PROBLEMS

Mustafa Servet Kiran and Mesut Gündüz

Computer Engineering Department
Selcuk University
Konya 42075, Turkey
{ mskiran; mgunduz }@selcuk.edu.tr

Received June 2011; revised October 2011

Abstract. *Artificial Bee Colony (ABC) is one of the popular algorithms of swarm intelligence. The ABC algorithm simulates foraging and dance behaviors of real honey bee colonies. It has high performance and success for numerical benchmark optimization problems. Although solution exploration of ABC algorithm is good, exploitation to found food sources is poor. In this study, inspiring Genetic Algorithm (GA), we proposed a crossover operation-based neighbor selection technique for information sharing in the hive. Local search and exploitation abilities of the ABC were herewith improved. The experimental results show that the improved ABC algorithm generates the solutions that are significantly more closed to minimal ones than the basic ABC algorithm on the numerical optimization problems and estimation of energy demand problem.*
Keywords: Swarm intelligence, Artificial bee colony, Numerical optimization, Crossover operation, Neighbor selection, Estimation of energy demand

1. **Introduction.** By now, there are many intelligent optimization techniques proposed by the researchers and their derivatives in the literature [1-7]. One of them is artificial bee colony (ABC) algorithm and it was developed by Karaboga, inspiring foraging and waggle dance behaviors of honey bee colonies [8]. Experimental results of works in literature show that ABC is better than other bio-inspiring algorithms such as genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE) [9-14].

Since 2005 year when ABC was proposed, various discrete and continuous ABC models have been developed. Many of these models are widely used in such a fields as design of digital IIR filters [15], leaf-constrained minimum spanning tree [16], for determining the sectionalizing switch to be operated in order to solve the distribution system loss minimization problem [17], quadratic minimum spanning tree [18], real parameter optimization [19], lot-streaming flow shop scheduling problem [20], constrained optimization problems [21], training neural networks [22], software test suite optimization [23], solving reliability redundancy allocation problems [24].

One of the important activities for honey bees is sharing the information about the positions of food sources around the hive. This activity is simply named as *information sharing*. Unfortunately, the existing ABC models cannot take into consideration this activity sufficiently and this situation has caused ineffective results for the forecasting problems according to our experiences. In this study, for improving the information sharing ability of the basic ABC model, the crossover operation between employed bees is proposed for producing neighbor bee of onlooker bees.

Onlooker bees in the basic ABC select employed bees in order to improve their solution. This selection mechanism is done according to quality of the solution found by employed bees. Briefly, the higher quality results the higher probability of selection. However, selection of neighbor employed bee and parameter for new candidate solutions is fully random. Inspiring GA [25-28], we proposed the crossover operation in order to produce neighbor bees for onlooker bees. Thus, onlooker bees use the information about position of food sources in the hive both selection an employed bee and the neighbor bee selection.

The rest of this study is organized as follows. The basic ABC algorithm is presented in Section 2 and the crossover operation and the crossover operators are explained in Section 3. The proposed model for ABC is elaborated in Section 4 and Section 5 presents experimental results. In order to show performance of proposed model, a real-world application is presented Section 6. Finally, conclusion and future works are given in Section 7.

2. **Overview of the ABC Algorithm.**

There are three kinds of honey bee in ABC to forage food source. They are employed bees, scout bees and onlooker bees. The tasks of these bees are to collect nectar around the hive, to random search for the nectar sources and to find the food sources based on the information received from the employed bees waggle dance. In ABC, food searching and nectar foraging around the hive are performed by scout, onlooker and employed bees collectively [13,14]. The algorithm of the simulation of foraging and dance behaviors of honey bee colony adopted from [8-14] is given in Figure 1.

In the algorithm ABC given in Figure 1, for generating an initial solution for ith employed bee Equation (1) is used.

$$x_i^j = x_{\min}^j + rand[0,1] * (x_{\max}^j - x_{\min}^j), \quad i = 1, 2, \ldots, N \text{ and } j = 1, 2, \ldots, D \tag{1}$$

where x_i^j is a parameter to be optimized for the ith employed bee on the dimension j of the D-dimensional solution space, N is the number of employed bees and $x_{\max}^j$ and $x_{\min}^j$ are the upper and lower bounds for x_i^j, respectively.

In both onlooker bee and employed bee phases, the food positions in the jth dimension are obtained by Equation (2).

$$v_{i,j} = x_{i,j} + \Phi(x_{i,j} - x_{k,j}) \quad j \in \{1, 2, \ldots, D\}, \quad k \neq i \text{ and } i, k \in \{1, 2, \ldots, N\} \tag{2}$$

where $x_{i,j}$ is ith employed bee, $v_{i,j}$ is the new solution for $x_{i,j}$ in the jth dimension, $x_{k,j}$ is a neighbor bee of $x_{i,j}$ in employed bee population. Here Φ is a number randomly selected in the range of $[-1, 1]$, D is the dimension of the problem, N is number of the employed bees, and $j \in \{1, 2, \ldots, D\}$ and $k \in \{1, 2, \ldots, N\}$ are selected randomly.

In order to generate a new food position, every onlooker bee memorizes the solution of one of n employed bees based on fitness values of the employed bees. The probability p_i of that an onlooker bee will select the selection of the solution of the ith employed bee is obtained as follows:

$$p_i = \frac{fit_i}{\sum_{j=1}^{n} fit_j} \tag{3}$$

where fit_i is the fitness value of ith employed bee and obtained as follows:

$$fit_i = \begin{cases} \dfrac{1}{1 + f_i} & \text{if } (f_i \geq 0) \\ 1 + abs(f_i) & \text{if } (f_i < 0) \end{cases} \tag{4}$$

where f_i is the objective function specific for the problem.

Initialization
For each employed bee
Generate initial solution for each employed bee by using Equation (1)
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
Reset the abandonment counter.
Until a termination condition is met {
Employed Bee Phase
For each employed bee
Select a neighbor employed bee randomly.
Update position of employed bee by using Equation (2) and position of neighbor bee.
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
If the fitness value of the new solution is better than the fitness value of the old solution then replace the old solution with new one and reset the abandonment counter of the new solution, else increase the abandonment counter of the old solution by 1
Onlooker Bee Phase
For each onlooker bee
Select an employed bee as neighbor randomly.
Select an employed bee for improvement its solution by using Equation (3)
Improve the solution of the employed bee by using Equation (2) and the neighbor
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
If the fitness value of the new solution is better than the fitness value of the old solution then replace the old solution with new one and reset the abandonment counter of the new solution, else increase the abandonment counter of the old solution by 1
Scout Bee Phase
Fix the abandonment counter H with the highest content
IF the content of the counter H is higher than the predefined limit **THEN** reset the counter H and by using Equation (1) generate a new solution for the employed bee to which the counter H belongs, **ELSE** continue
} **End Until**

FIGURE 1. The ABC algorithm

In addition, Equation (1) is used in order to generate new solution for the scout bee in the scout bee phase of the ABC and all the onlooker bees use Equation (2) so as to improve the solution.

3. The Crossover Operation.

In genetic algorithms, crossover operation is used for producing a new generation from parents. Firstly, the members with good solution of the population placed into mating pool. Then, crossover operation is applied to chromosomes of two parents selected from mating pool and changes mutually defined parts of the two parents. A lot of different crossover operators were proposed in literature [29-33] but one point, two point, multi-point and uniform operators are widely used for crossover operation.

3.1. The one-point crossover operator.

Only one crossover point is randomly selected between 1 and D-1 where D is the length of the chromosomes. Each chromosome is divided into two parts as left of the point and right of the point. Right part of the first chromosome are replaced right part of the second chromosome. Thus, two offspring are obtained from two parents. The one-point crossover operation is showed in Figure 2.

3.2. The two-point crossover operator.

In two point crossover operator, two different points between 1 and D-1 randomly selected divide the chromosome into three parts.

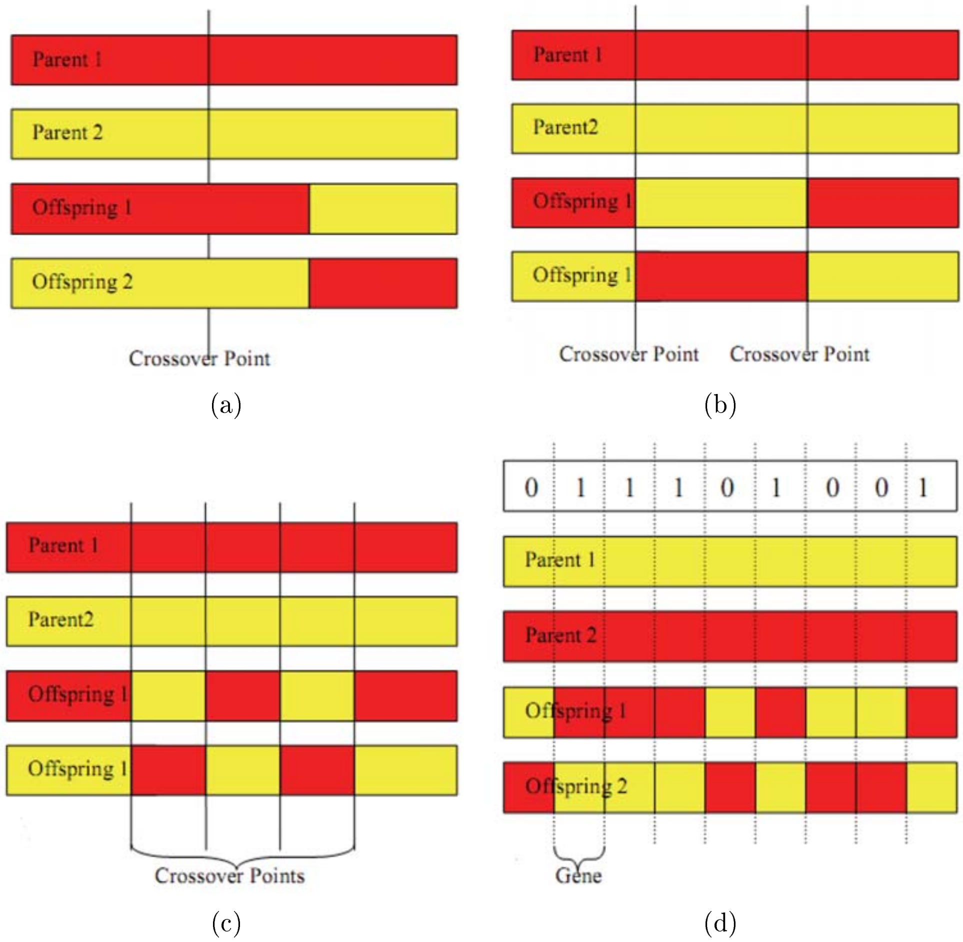

FIGURE 2. Crossover operators. (a) One-point, (b) two-point, (c) multi-point and (d) uniform.

The middle parts of the chromosomes are replaced and two offspring is produced. This situation is showed in Figure 2.

3.3. **The multi-point crossover operator.** Many different cut off points are randomly selected and chromosomes are divided by these points. By being replaced parts of chromosomes, offspring are obtained. The multi-point crossover operation is showed in Figure 2.

3.4. **The uniform crossover operator.** Uniform crossover operator is applied to parent chromosomes in gene level. Firstly, a temporary bit string having the same length with the chromosomes is created. If nth index of temporary string is 0, offspring 1 takes gene from Parent 1 and offspring 2 take gene from Parent 2 otherwise offspring 1 takes gene from Parent 2 and offspring 2 takes gene from Parent 1. In uniform crossover operator, crossover points are randomly selected. The uniform crossover operator is given in Figure 2.

4. The Proposed Model for ABC.

4.1. **Using crossover operation for neighbor selection of the onlooker bees.** In this study, the crossover operation is used for information sharing between employed and onlooker bees in the hive. The onlooker bees learn locations of food sources by watching the waggle dance of employed bees in dance area reserved in the hive for this aim. Remember that in the basic ABC algorithm, the onlooker bees do not use this information for the neighbor selection. Every onlooker bee selects an employed bee as neighbor for itself randomly. In this paper, we propose a crossover operation-based model for selection of neighbor for an onlooker bee. In our model, a certain number of employed bees with the good solutions are put into the mating pool. The collection of bees is named as parents. Then, the solution parameters of the parents are subjected to the crossover operation. The best offspring obtained as result of this operation is considered as neighbor for onlooker bees. The new solution in the onlooker bee phase of the ABC is obtained by Equation (5), which is a modified form of Equation (2) for onlooker bees.

$$v_{i,j} = x_{i,j} + \Phi(x_{i,j} - B_j) \tag{5}$$

where $x_{i,j}$ is the jth parameter of the ith employed bee, $v_{i,j}$ is candidate solution for jth parameter of ith onlooker bee and B_j is the jth parameter of the best offspring.

4.2. **The ABC algorithm with the crossover operation.** In order to benefit from information about positions of food sources given by the employed bees, we improved the ABC by adding to it the crossover operation as a new phase for this algorithm. In this phase, a given number of employed bees (parents) with solutions better than others are subjected to crossover operation. As result of this operation the best offspring to be considered as the neighbor for the onlooker bee is obtained. Since by the crossover operation produces the neighbor bee for the onlooker bee more exactly than this is done by the random selection for each onlooker bee, the proposed algorithm will produce the more accurate results. The ABC algorithm with the mentioned crossover operation, referred as CABC, is given in Figure 3.

4.3. **The estimation of the time complexity of the CABC algorithm.** As it is seen from Figure 3, the algorithm CABC consists of four sequentially realized phases, named as Initialization phase (IP), Employed bee phase (EBP), Onlooker bee phase (OBP) and Scout bee phase (SBP).

There are two nested loops in the IP the external from which iterates N (the number of employed bees) times and the internal one iterates D (the dimensionality of the solution space) times in each of iterations of the external loop. Therefore, the time complexity of this phase is W(IP) = O($N \times D$). Since always D is less or equal to N, the worst case time complexity of this phase is to be W(IP) = O(N^2).

The EBP is a While loop with a single For loop in it. The While loop iterates at most 5000 times and the For loop iterates N times in each of its iterations. Therefore, the worst case time complexity of this phase is W(EBP) = O(5000×N). But since the worst case value of N is 50, W(EBP) = O ($2 \times N^2 \times N$) = O(N^3).

The OBP differs from the EBP in that it performs the crossover operation in each of iterations of the For loop. Since this operation is performed by the quicksort algorithm with the time complexity of O(N^2), the worst case time complexity of this phase is to be W(OBP) = O(N^5).

The SBP is identical to EBP with respect to numbers of iterations of the While and For loops. Therefore, its worst case time complexity is to be W(SBP) = W(EBP) = O(N^3).

According to [34,35], the worst case time complexity of the algorithm CABC given in Figure 3 is to be W(CABC) = max{W(IP), W(EBP), W(OBP), W(SBP)} = W(OBP)= O(N^5).

5. **The Experimental Estimation of the Proposed Algorithm.** The experiments were performed by using a personal computer with a 3.04GHz Intel Core 2 Duo E8400 Pentium microprocessor, 2GB RAM and Windows XP OS and Matlab v.7.0.4 platform. In

Initialization
For each employed bee
Generate initial solution for each employed bee by using Equation (1)
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
Reset the abandonment counter.
Until a termination condition is met {
Employed Bee Phase
For each employed bee
Select a neighbor employed bee randomly.
Update position of employed bee by using Equation (2) and position of neighbor bee.
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
IF the fitness value of the new solution is better than the fitness value of the old solution **THEN** replace the old solution with new one and reset the abandonment counter of the new solution, **ELSE** increase the abandonment counter of the old solution by 1
Onlooker Bee Phase
For each onlooker bee
Select an employed bee according to fitness values of employed bees
Crossover Operation
Sort the fitness values of employed bees in descending order.
Put *n* first employed bees (parents) into mating pool.
Subject all parents present in mating pool to the crossover operation
Pick all offspring produced as result of previous step
Calculate the objective function values for each offspring by the equation specific for the problem being solved
Calculate the fitness values (*fit*) for offspring by using Equation (4)
Determine the neighbor with the best fitness value.
Improve the solution of employed bee selected by using Equation (2) and the neighbor.
Calculate the objective function values by the equation specific for the problem
Calculate the fitness value (*fit*) by using Equation (4)
IF the fitness value of the new solution is better than the fitness value of the old solution **THEN** replace the old solution with new one and reset the abandonment counter of the new solution, **ELSE** increase the abandonment counter of the old solution by 1
Scout Bee Phase
Fix the abandonment counter H with the highest content
IF the content of the counter H is higher than the predefined limit **THEN** reset the counter H and by using Equation (1) generate a new solution for the employed bee to which the counter H belongs, **ELSE** continue
} **End Until**

FIGURE 3. The CABC algorithm

the experiments, four kinds of crossover operators (one-point, two-point, multi-point and uniform) were used. While to 2-D benchmark functions were applied the only one-point crossover operator, to the other benchmark functions all mentioned crossover operators were applied. As the maximum evaluation number (MEN) of the function being processed the 10000 times the dimensionality of solution space of that function was used.

5.1. **The benchmark functions used in the experiments.** In order to test the performance of the proposed CABC algorithm, in the experiments, widely used 10 numerical benchmark functions [8-14,23] given in Table 1 were used.

In Table 1, D column is the dimension of functions, C column is the characteristic of the functions. If a function has more than one local optimum, this function is called as multimodal (6 hump camel back, booth, schaffer, rastrigin, schwefel) otherwise functions are unimodal (matyas, trid, sumsquares, sphere, rosenbrock). If $F(x, y)$ function can

TABLE 1. The benchmark functions used in the experiments

Name	D	C*	Range	Formulation	Minimum
f_1 Six Hump Camel Back	2	MN	$[-5,5]$	$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	-1.03163
f_2 Matyas	2	UN	$[-10,10]$	$f(x)=0.26(x_1^2+x_2^2)-0.48(x_1x_2)$	0
f_3 Booth	2	MS	$[-10,10]$	$f(x)=(x_1+2x_2-7)^2+(2x_1+x_2-5)^2$	0
f_4 Schaffer	2	MN	$[-100,100]$	$f(x)=0.5+\frac{\sin^2\left(\sqrt{x_1^2+x_2^2}\right)-0.5}{(1+0.001(x_1^2+x_1^2))^2}$	0
f_5 Trid	10	UN	$[-D^2,D^2]$	$f(x)=\sum_{i=1}^{D}(x_i-1)^2-\sum_{i=2}^{D}x_ix_{i-1}$	-210
f_6 SumSquares	10 30 50	US	$[-10,10]$	$f(x)=\sum_{i=1}^{D}ix_i^2$	0
f_7 Sphere	10 30 50	US	$[-100,100]$	$f(x)=\sum_{i=1}^{D}x_i^2$	0
f_8 Rastrigin	10 30 50	MS	$[-5.12,5.12]$	$f(x)=\sum_{i=1}^{D}[x_i^2-10\cos(2\pi x_i)+10]$	0
f_9 Schwefel	10 30 50	MS	$[-500,500]$	$f(x)=\sum_{i=1}^{D}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	$-418.9829\times D$
f_{10} Rosenbrock	10 30 50	UN	$[-30,30]$	$f(x)=\sum_{i=1}^{D-1}\left[100\left(x_{i+1}-x_i^2\right)^2+(x_i-1)^2\right]$	0

* U: Unimodal, M: Multimodal, S: Separable, N: Non-separable

be written in the form of $F(x,y)=f(x)+g(y)$, these functions are separable (booth, sumsquares, sphere, rastrigin, schwefel). Non-separable functions are note written in this form (Six hump camel back, matyas, schaffer, trid, rosenbrock). Unimodal functions tests exploitation ability of the algorithm and search and getting rid of local minimum abilities of the algorithm were tested by multimodal functions. Moreover, all the benchmark functions used in the experiments were illustrated in Figure 4.

5.2. **Parameter settings of ABC and CABC.** In all experiments, the population size was taken as 100. Half of the population consists of the employed bees, the other half contains onlooker bees. In a certain time, only one scout bee can be in the population. Control parameter of the scout bee (limit), which is used for abandonment when the solution of the employed bee cannot be improved, was calculated as follows:

$$limit = D \times N \tag{6}$$

where D is the dimension of function and N is the number of employed bees. In the CABC, the mating pool size has been kept as $0.1\times$ number of employed bees (in our experiments, number of employed bees is 50 so mating pool size is 5) because our purpose is to produce a neighbor, not to produce a generation. When to choose high number for the mating pool size, the neighbor bee cannot be improved because the parents with bad solution include in the mating pool. When to choose small value for the mating pool size, the bee population show the stagnation behavior and the good solutions cannot be obtained. Also, number of crossover points in the multipoint and uniform crossover operators was taken as 5 and the crossover points were selected randomly.

5.3. **Experimental results.** For all benchmark functions, ABC and CABC algorithms were repeated 30 times independently. Objective function values less than E-20 were

Table 2. The results obtained by the basic ABC and CABC algorithms

Function			Basic ABC		$ABC_{one\text{-}point}$		$ABC_{two\text{-}point}$		$ABC_{multi\text{-}point}$		$ABC_{uniform}$	
No	D	Minimum	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
f_1	2	−1.03163	−1.03163	0	−1.03163	0	–	–	–	–	–	–
f_2	2	0	0.00012	0.0001	**5.1875E-05**	**7.35E-05**	–	–	–	–	–	–
f_3	2	0	3.2940E-011	4.9863e-011	**3.95546E-14**	**6.81E-14**	–	–	–	–	–	–
f_4	2	0	8.0215E-005	1.8089e-004	**7.13537e-005**	**2.1375e-004**	–	–	–	–	–	–
f_5	10	−210	−209.7713	0.1647	−209.7836	0.50486	−209.7522	0.3643	−209.8127	0.4267	**−209.9252**	**0.0776**
f_6	10	0	7.8859E-017	1.6974e-017	8.2365E-17	1.46E-17	**7.7434E-17**	**1.6E-17**	7.84018E-17	1.4076E-17	7.9762e-017	1.5707e-017
	30		4.9112E-016	4.5668e-017	4.65927E-16	7.5E-17	**4.21943E-16**	**8.81E-17**	4.39821E-16	8.246E-17	4.6632e-016	6.9162e-017
	50		9.5926E-016	1.2270e-016	8.6846E-16	9.55E-17	8.7507E-16	1.06E-16	**8.49426E-16**	**1.13586E-16**	8.7560e-016	1.1205e-016
f_7	10	0	**7.8473E-017**	**1.8762e-017**	8.10139E-17	1.48E-17	8.69701E-17	1.96E-17	8.29788E-17	1.14028E-17	8.3756e-017	2.8684e-017
	30		4.7485E-016	5.9175e-017	4.62407E-16	8.8E-17	4.54905E-16	7.69E-17	**4.49752E-16**	**8.58257E-17**	4.7284e-016	5.5235e-017
	50		9.5672E-016	1.1039e-016	8.9441e-016	1.2255e-016	8.33328E-16	1.07E-16	8.71083E-16	1.16502E-16	**8.2782e-016**	**9.97E-016**
f_8	10	0	0	0	0	0	0	0	0	0	0	0
	30		0	0	0	0	0	0	0	0	0	0
	50		0	0	0	0	0	0	0	0	0	0
f_9	10	−4189.829	−4189.8289	0	−4189.8289	0	−4189.8289	0	−4189.8289	0	−4189.8289	0
	30	−12569.487	−12569.4866	0	−12569.4866	0	−12569.4866	0	−12569.4866	0	−12569.4866	0
	50	−20949.145	−20949.1444	0	−20949.1444	0	−20949.1444	0	−20949.1444	0	−20949.1444	0
f_{10}	10	0	0.0428	0.0535	0.0470	0.0899	0.0473	0.0789	**0.0240**	**0.0314**	0.0379	0.0648
	30		0.0343	0.0409	0.0367	0.0421	0.0838	0.1374	0.0584	0.1299	**0.0285**	**0.05157**
	50		0.0656	0.1132	0.0699	0.1006	0.0982	0.1290	0.0842	0.1271	**0.0539**	**0.0933**

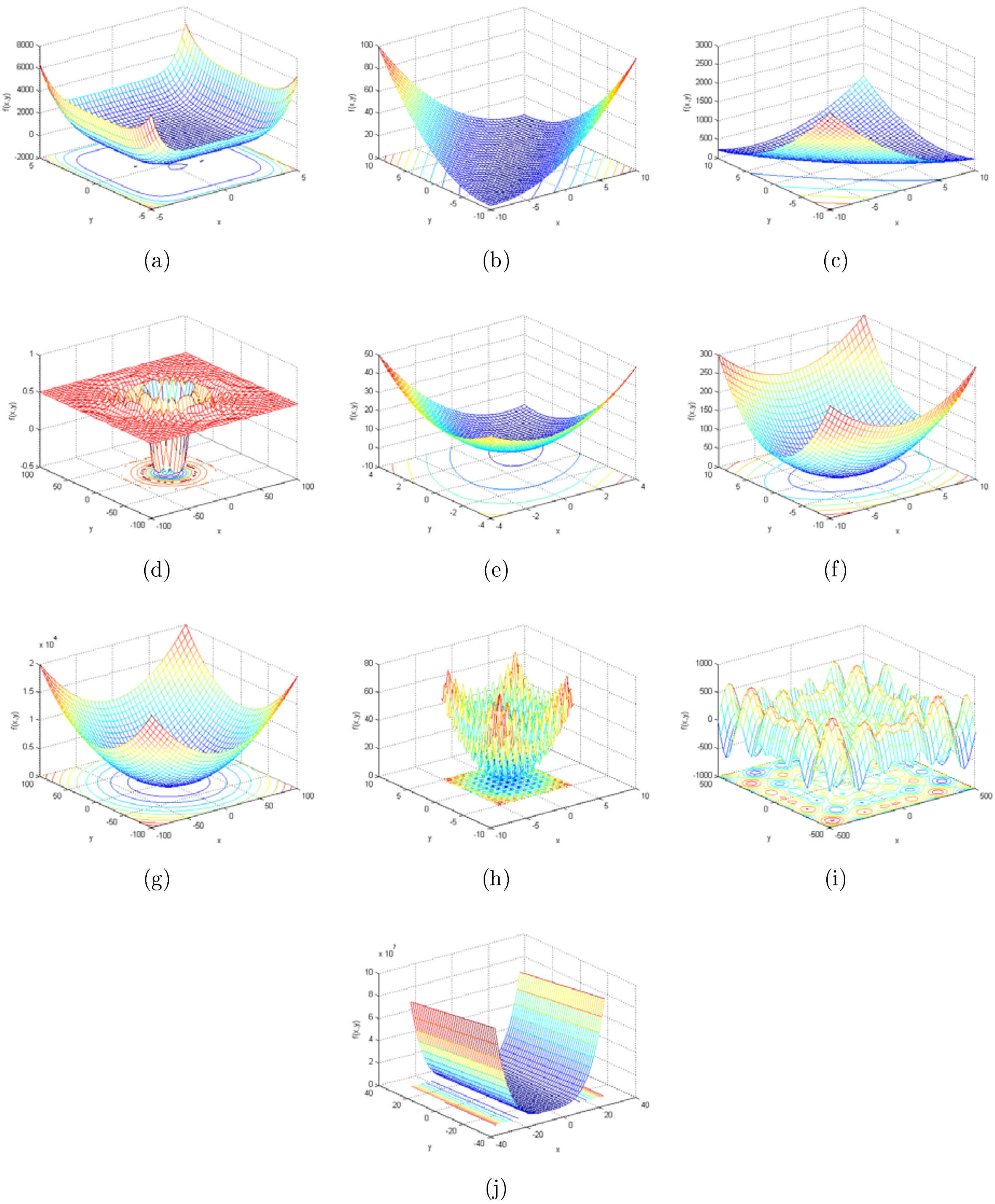

FIGURE 4. Surface plots of the benchmark functions. (a) Six hump camel back, (b) matyas, (c) booth, (d) schaffer, (e) trid, (f) sumsquares functions, (g) sphere, (h) rastrigin, (i) schwefel, (j) rosenbrock functions.

reported as 0. If the dimension of the function equals to two, CABC with one-point crossover operator is applied to the function.

For clearly comparison of ABC and CABC algorithms, the results obtained by CABCs with crossover operators and basic ABC are given in the same table (Table 2) and the best result obtained for each function is written in the bold type.

Considering Table 2, it is seen that the ABC and CABC achieved to global minimum on the f_1 and f_8 functions. On the f_7 function ($D = 10$), ABC produced better result than CABC. On the f_2, f_3 and f_4 functions, $\text{CABC}_{\text{one-point}}$ is better than ABC. $\text{CABC}_{\text{two-point}}$ is better than the other methods on the f_6 function ($D = 10$ and $D = 30$). $\text{CABC}_{\text{multi-point}}$ showed better performance than the other methods on the f_6 ($D = 50$), f_7 ($D = 30$) and f_{10} ($D = 10$) functions. Finally, $\text{CABC}_{\text{uniform}}$ showed better performance than ABC on the f_5, f_7 ($D = 50$) and f_{10} ($D = 30$ and $D = 50$) functions. Consequently, CABC algorithms produced better results than the basic ABC algorithm on the benchmark functions.

In addition, running times of the basic ABC and CABCs were analyzed. The worst running times of the algorithms were given in Table 3. The elapsed times reported in Table 3 are the worst running times obtained from the 30 independent runs.

Table 3. The worst running times obtained by 30 runs

Function	**D**	**Elapsed Times (seconds)***				
		Basic ABC	$\text{ABC}_{\text{one-point}}$	$\text{ABC}_{\text{two-point}}$	$\text{ABC}_{\text{multi-point}}$	$\text{ABC}_{\text{uniform}}$
f_1 (Six Hump Camel Back)	2	1.3948	1.8534	–	–	–
f_2 (Matyas)	2	1.4125	1.76211	–	–	–
f_3 (Booth)	2	1.4187	1.7656	–	–	–
f_4 (Schaffer)	2	1.4564	1.8731	–	–	–
f_5 (Trid)	10	7.7739	9.6446	10.5986	10.9867	10.9577
f_6 (SumSquares)	50	38.8197	53.5744	53.7117	55.4224	54.8307
f_7 (Sphere)	50	37.9113	51.5048	52.0082	54.1913	53.4573
f_8 (Rastrigin)	50	38.4297	57.3003	58.3818	63.3698	59.2113
f_9 (Schwefel)	50	38.4641	69.6321	72.6918	74.7356	74.7984
f_{10} (Rosenbrock)	50	38.5371	56.3754	56.7983	57.7755	56.6189

* tic and toc functions of Matlab v.7.0.4 were used for calculating elapsed times for the runs

As seen from Table 3, CABC algorithms are slower than the ABC algorithm because there is crossover operation in the CABC but remarkable solutions are obtained in a reasonable time by using CABC algorithms.

6. **A Real World Application – Estimation of Energy Demand Problem.** Energy, as a resource of many things, has importance for sustaining production in developing countries. If the countries know how much energy needs in the future, they use current energy resources effectively, and they investigate new energy sources (such as renewable energy) in order to meet self demands. Therefore, forecasting energy demand is significant problem for researchers and policy makers. Energy estimation demand depends on many socio-economic factors such as gross domestic products (GDP), population, import, export, growth rate and availability energy resources [36] but energy demand of a country is mostly affected by population, GDP, import and export indicators [36-38].

In order to show performance and success of CABC algorithms, we used the estimation energy demand problem of Turkey. While GDP, population, import and export increased 3.4, 0.63, 22 and 31.5 times respectively, energy consumption of Turkey has increased 1.98 times between 1979 and 2005 years (Table 4 taken from [39,40]), and this increasing show us that there are a linear relationship between the indicators and energy consumption. Thus, model of forecasting demand based on socio-economic indicators was modeled by using linear form expressed in Equation (7).

$$E_{linear} = w_1 \cdot X_1 + w_2 \cdot X_2 + w_3 \cdot X_3 + w_4 \cdot X_4 + w_5 \quad (7)$$

TABLE 4. Energy demand, GDP, population, import, export data of Turkey

Year	Energy Demand (MTOE)	GDP ($\$10^9$)	Population (10^6)	Import ($\$10^9$)	Export ($\$10^9$)
1979	30.71	82.00	43.53	5.07	2.26
1980	31.97	68.00	44.44	7.91	2.91
1981	32.05	72.00	45.54	8.93	4.70
1982	34.39	64.00	46.69	8.84	5.75
1983	35.70	60.00	47.86	9.24	5.73
1984	37.43	59.00	49.07	10.76	7.13
1985	39.40	67.00	50.31	11.34	7.95
1986	42.47	75.00	51.43	11.10	7.46
1987	46.88	86.00	52.56	14.16	10.19
1988	47.91	90.00	53.72	14.34	11.66
1989	50.71	108.00	54.89	15.79	11.62
1990	52.98	151.00	56.10	22.30	12.96
1991	54.27	150.00	57.19	21.05	13.59
1992	56.68	158.00	58.25	22.87	14.72
1993	60.26	179.00	59.32	29.43	15.35
1994	59.12	132.00	60.42	23.27	18.11
1995	63.68	170.00	61.53	35.71	21.64
1996	69.86	184.00	62.67	43.63	23.22
1997	73.78	192.00	63.82	48.56	26.26
1998	74.71	207.00	65.00	45.92	26.97
1999	76.77	187.00	66.43	40.67	26.59
2000	80.50	200.00	67.42	54.50	27.78
2001	75.40	146.00	68.37	41.40	31.33
2002	78.33	181.00	69.30	51.55	36.06
2003	83.84	239.00	70.23	69.34	47.25
2004	87.82	299.00	71.15	97.54	63.17
2005	91.58	361.00	72.97	116.77	73.48

CABC estimation model tries to optimize coefficients (w_i) of design parameters (X_i). In energy demand estimation, we aim to find the fittest model to the data. The objective function of the model is given by Equation (8).

$$\min f(v) = \sum_{k=1}^{R} \left(E_k^{observed} - E_k^{predicted} \right)^2 \tag{8}$$

where $E^{observed}$ and $E^{predicted}$ are the actual and predicted energy demand, respectively and k is the observations.

For finding optimum set of w and setting E_{linear} equation, the basic ABC and CABC algorithm were applied. The population size, limit value and maximum cycle number used as stopping criterion for the algorithms were chosen as 100, 500 and 5000, respectively. Also, all the solutions' parameters of employed and onlooker bees were updated at each iteration in order to increase exploitation food sources found.

Coefficients in the linear form and objective functions values obtained by the algorithms were given in below.

Obtained by the basic ABC algorithm:

$$\begin{aligned} E_{linear} &= 0.065622 \cdot X_1 + 1.72044 \cdot X_2 + 0.212867 \cdot X_3 - 0.41529 \cdot X_4 - 50.4559 \\ f(v) &= 84.31805 \end{aligned} \tag{9}$$

Obtained by $\text{CABC}_{\text{one-point}}$ algorithm:

$$\begin{aligned} E_{linear} &= 0.003759 \cdot X_1 + 1.912559 \cdot X_2 + 0.373371 \cdot X_3 - 0.48326 \cdot X_4 - 55.9091 \\ f(v) &= 41.70295 \end{aligned} \tag{10}$$

Obtained by $\text{CABC}_{\text{two-point}}$ algorithm:

$$\begin{aligned} E_{linear} &= 0.003762 \cdot X_1 + 1.912553 \cdot X_2 + 0.373361 \cdot X_3 - 0.48325 \cdot X_4 - 55.9091 \\ f(v) &= 41.70295 \end{aligned} \tag{11}$$

Obtained by $\text{CABC}_{\text{multi-point}}$ algorithm:

$$\begin{aligned} E_{linear} &= 0.003756 \cdot X_1 + 1.912578 \cdot X_2 + 0.373404 \cdot X_3 - 0.4833 \cdot X_4 - 55.9099 \\ f(v) &= 41.70295 \end{aligned} \tag{12}$$

Obtained by $\text{CABC}_{\text{uniform}}$ algorithm:

$$\begin{aligned} E_{linear} &= 0.003768 \cdot X_1 + 1.912523 \cdot X_2 + 0.373315 \cdot X_3 - 0.48319 \cdot X_4 - 55.908 \\ f(v) &= 41.70295 \end{aligned} \tag{13}$$

where X_1 is GDP, X_2 is population, X_3 is import, X_4 is export and $f(v)$ is sum of the squared errors. In addition, the performances of the models were validated by using coefficients mentioned above on the 1996-2005 data. Table 5 shows relative errors between estimated and observed data. According to Table 5, $\text{CABC}_{\text{one-point}}$ has lower relative error rates than the basic ABC algorithm.

TABLE 5. Validation of the models

Year	Observed Energy Demand (MTOE)	Basic ABC		$\text{ABC}_{\text{one-point}}$	
		Estimation Energy Demand (MTOE)	Relative Error (%)	Estimation Energy Demand (MTOE)	Relative Error (%)
1996	69.86	69,08	−1,12	**69,71**	**−0,21**
1997	73.78	71,37	−3,27	**72,31**	**−1,99**
1998	74.71	**73,53**	**−1,58**	73,3	−1,89
1999	76.77	73,72	−3,97	**74,18**	**−3,37**
2000	80.50	78,73	−2,2	**80,71**	**0,26**
2001	75.40	72,55	−3,78	**75,72**	**0,42**
2002	78.33	76,65	−2,14	**79,13**	**1,02**
2003	83.84	81,19	−3,16	**82,36**	**−1,77**
2004	87.82	86,1	−1,96	**87,18**	**−0,73**
2005	91.58	93,12	1,68	**93,1**	**1,66**

As seen from objective function values ($f(v)$) mentioned above, CABC algorithms produced better coefficients set than the basic ABC algorithm. The reason for this is the information sharing and that the onlooker bees more affected by the best solutions found by the employed bees. The objective function values obtained by the CABC algorithms are the same but the convergence to the minimum which is shown in Figure 5 are different. As there are fewer dimensions of the problem (number of coefficients), convergence rate of $\text{CABC}_{\text{one-point}}$ is better than the others.

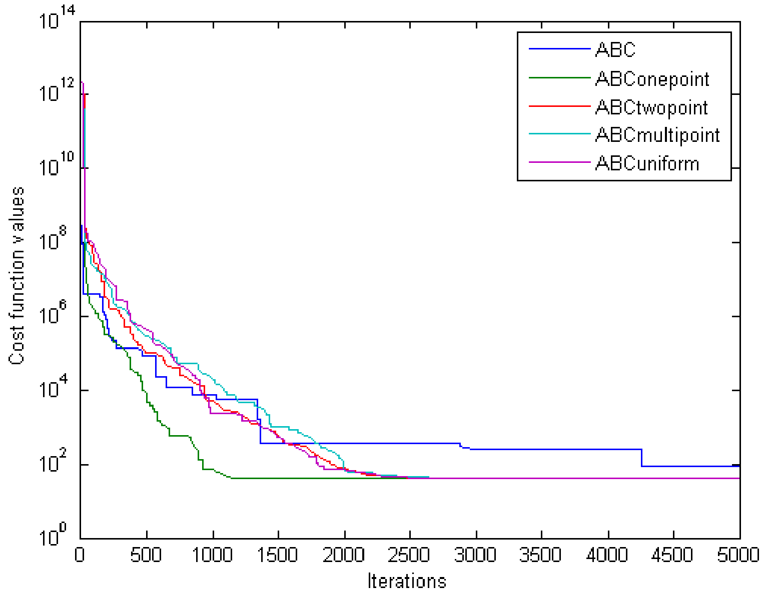

FIGURE 5. Convergence graph of the algorithms

Future projections of energy demand can be drawn under the different scenarios by using coefficients found. Kıran et al. discussed the estimation of energy demand for Turkey detailed in [36]. Therefore, these scenarios were not given in this paper.

7. **Conclusion and Future Works.** The information sharing is very important argument for swarm intelligence algorithms. Although exploration solution space of basic ABC algorithm is good, exploitation to food source found is insufficient because the information in the hive is not used enough. In this study, the method proposed by inspiring GA for the information sharing produced the results better than the basic ABC on the well-known numeric optimization problems and the estimation of energy demand. Future works include implementation of CABC to different optimization problems and to forecast CO2 emission of Turkey.

Acknowledgement. The authors thank to Dervis KARABOGA for suggesting different crossover operators, Sirzat KAHRAMANLI for corrections in writing and analysis of time complexity of the algorithm, Eren Özceylan for explaining the estimation energy demand problem and anonymous reviewers for their valuable comments and contributions. This study has been supported by Selcuk University Coordinatorship of Scientific Research Projects.

REFERENCES

[1] M. Dorigo and M. Gambartella, Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computing*, vol.26, no.1, pp.53-66, 1997.
[2] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proc. of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, pp.1942-1948, 1995.
[3] S.-C. Chu and P.-W. Tsai, Computational intelligence based on the behaviors of cats, *International Journal of Innovative Computing, Information and Control*, vol.3, no.1, pp.163-173, 2007.
[4] X. Cai, Z. Cui, J. Zeng and Y. Tan, Particle swarm optimization with self-adjusting cognitive selection strategy, *International Journal of Innovative Computing, Information and Control*, vol.4, no.4, pp.943-952, 2008.
[5] F. Kang, J. Lie and Z. Ma, Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions, *Information Science*, vol.181, no.6, pp.3508-3531, 2011.

[6] Z. Cui, J. Zeng and G. Sun, A fast particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.2, no.6, pp.1365-1380, 2006.
[7] Y. Guo, X. Gao, H. Yin and Z. Tang, Coevolutionary optimization algorithm with dynamic subpopulation size, *International Journal of Innovative Computing, Information and Control*, vol.3, no.2, pp.435-448, 2007.
[8] D. Karaboga, An idea based on honey bee swarm for numerical optimization, *Technical Report-TR06*, Erciyes University, Kayseri/Turkey, 2005.
[9] B. Basturk and D. Karaboga, An artificial bee colony (ABC) algorithm for numeric function optimization, *IEEE Swarm Intelligence Symposium*, Indianapolis, IN, USA, 2006.
[10] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, vol.39, no.3, pp.459-471, 2007.
[11] D. Karaboga and B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, vol.8, no.1, pp.687-697, 2008.
[12] D. Karaboga and B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, vol.214, no.1, pp.108-132, 2009.
[13] G. Zhu and S. Kwong, Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied Mathematics and Computation*, vol.217, no.7, pp.3166-3173, 2010.
[14] P.-W. Tsai, J.-S. Pan, B.-Y. Liao and S.-C. Chu, Enhanced artificial bee colony optimization, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(B), pp.5081-5092, 2009.
[15] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute*, vol.346, no.4, pp.328-348, 2009.
[16] A. Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, *Applied Soft Computing*, vol.9, no.2, pp.625-631, 2009.
[17] R. S. Rao, S. Narasimham and M. Ramalingaraju, Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm, *International Journal of Electrical and Electronics Engineering*, vol.2, no.10, pp.644-650, 2008.
[18] S. Sundar and A. Singh, A swarm intelligence approach to the quadratic minimum spanning tree problem, *Information Sciences*, vol.180, no.17, pp.3182-3191, 2010.
[19] B. Akay and D. Karaboga, A modified artificial bee colony algorithm for real-parameter optimization, *Information Science*, 2010.
[20] Q. K. Pan, M. F. Tasgetiren, P. N. Suganthan and T. J. Chua, A discrete artificial bee colony algorithm for the lot-streaming flow shop scheduling problem, *Information Sciences*, vol.181, no.12, pp.2455-2468, 2011.
[21] D. Karaboga and B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Lecture Notes in Computer Science, Advances in Soft Computing-Foundations of Fuzzy Logic and Soft Computing*, vol.4529, pp.789-798, 2007.
[22] D. Karaboga and C. Ozturk, Neural networks training by artificial bee colony algorithm on pattern classification, *Neural Network World*, vol.19, no.3, pp.279-292, 2009.
[23] D. J. Mala and V. Mohan, ABC tester – Artificial bee colony based software test suite optimization approach, *International Journal of Software Engineering*, vol.2, no.2, pp.15-43, 2009.
[24] W. C. Yeh and T. J. Hsieh, Solving reliability redundancy allocation problems using an artificial bee colony algorithm, *Computers & Operations Research*, vol.38, no.11, pp.1465-1473, 2011.
[25] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
[26] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, New York, 1989.
[27] A. E. Eiben, E. P. Raue and Z. Ruttkay, Genetic algorithms with multi-parent recombination, *Parallel Problem Solving from Nature – PPSN III, Lectures Notes in Computer Science*, vol.866, pp.78-87, 1994.
[28] K. Kim, M. Gen and M. Kim, Adaptive genetic algorithms for multi-resource constrained project scheduling problem with multiple models, *International Journal of Innovative Computing, Information and Control*, vol.2, no.1, pp.41-49, 2006.
[29] M. Kaya, The effects of two new crossover operators on genetic algorithm performance, *Applied Soft Computing*, vol.11, no.1, pp.881-890, 2011.
[30] K. Deep and T. Manij, A new crossover operator for real coded genetic algorithms, *Applied Mathematics and Computation*, vol.188, no.1, pp.895-911, 2007.

[31] C. G. Martinez, M. Lozano, D. Molina and A. M. Sanchez, Global and local real-coded genetic algorithms based on parent-centric crossover operators, *European Journal of Operational Research*, vol.185, no.3, pp.1088-1113, 2008.
[32] D. O. Boyer, C. H. Martinez and N. G. Pedrajas, A crossover operator for evolutionary algorithms based on population features, *Journal of Artificial Intelligence Research*, vol.24, pp.1-48, 2005.
[33] M. Watanabe and K. Ida, A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem, *Computers & Industrial Engineering*, vol.48, no.4, pp.743-752, 2005.
[34] E. Tomita, A. Tanaka and H. Takahashi, The worst-case time complexity for generating all maximal cliques and computational experiments, *Theoretical Computer Science*, vol.363, no.1, pp.28-42, 2006.
[35] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*, Cambridge University Press, New York, 2009.
[36] M. S. Kıran, E. Özceylan, M. Gündüz and T. Paksoy, A novel hybrid approach based on particle swarm optimization and ant colony algorithm to forecast energy demand of turkey, *Energy Conversion and Management*, vol.53, no.1, pp.75-83, 2012.
[37] A. Ünler, Improvement of energy demand forecast using swarm intelligence: The case of Turkey with projections to 2025, *Energy Policy*, vol.36, no.6, pp.1937-1944, 2008.
[38] M. D. Toksarı, Ant colony optimization approach to estimate energy demand of Turkey, *Energy Policy*, vol.35, no.8, pp.3984-3990, 2007.
[39] TSI, Turkish Statistical Institute, *http://www.tuik.gov.tr*, [accessed 22.09.2011].
[40] MENR, The Ministry of Energy and Natural Resources, *http://www.enerji.gov.tr*, [accessed 22.09.2011].