Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2014) 22: 1177 – 1192
© TÜBİTAK
doi:10.3906/elk-1211-171

Induction motor parameter estimation using metaheuristic methods

Ali İhsan ÇANAKOĞLU[1], **Asım Gökhan YETGİN**[1,*], **Hasan TEMURTAŞ**[2], **Mustafa TURAN**[3]

[1]Department of Electrical & Electronic Engineering, Faculty of Engineering, Dumlupinar University, Kütahya, Turkey
[2]Department of Computer Engineering, Faculty of Engineering, Dumlupinar University, Kütahya, Turkey
[3]Department of Electrical & Electronic Engineering, Faculty of Engineering, Sakarya University, Sakarya, Turkey

Received: 30.11.2012 • **Accepted:** 31.01.2013 • **Published Online:** 15.08.2014 • **Printed:** 12.09.2014

Abstract: The steady-state equivalent circuit parameters of an induction motor can be estimated using the operation characteristics that are provided by manufacturers. The characteristics of the motor used in estimation methods are the starting, maximum, and nominal torque values; the power factor; and efficiency. The operation characteristics of a motor given in data sheets are generally based on design parameters and are not suitable with real values. For this reason, in this paper, the data used in the parameter estimation for induction motors are taken from the literature. Using an optimization method for parameter estimation is useful for comparing the manufacturer values and values at the end of estimation, as well as minimizing the error in between. There are many methods in the literature for the parameter estimation of induction motors. In this study, the estimation is made using the charged system search (CSS), differential evolution algorithm (DEA), particle swarm optimization, and genetic algorithm optimization techniques. The CSS algorithm is first applied for estimation of the parameters of an induction motor. The results obtained from all of the methods show that the CSS algorithm is suitable with the DEA. From the obtained results, it is understood that an exact approach can be made to equivalent circuit parameters in case the values given by the manufacturer model the motor properly.

Key words: Induction motor, exact equivalent circuit parameters, torque values, charged system search, differential evolution algorithm, particle swarm optimization, genetic algorithm

1. Introduction

Induction motors have important problems, such as transient and quasi-steady-state stability. To solve the steady-state stability problems of an induction motor, equivalent circuit parameters are required. These parameters are the resistances and reactances of the stator and rotor, including magnetizing branches. Estimation of these parameters is particularly essential in determining their effects on motor performance. The main difficulty in constructing an accurate motor model is the unavailability of manufacturer data for estimation. Hence, explicit representation of induction motor models is not given in various applications. In the conventional techniques, estimation of the induction motor parameters is based on no-load and blocked-rotor tests [1].

Aside from the conventional technique, there are 2 different approaches for parameter estimation, which are online and offline techniques. The former uses the Kalman filter [2] and least square techniques [3]. The latter is offline [4] curve generation for the experimentally measured data. Recently, artificial neural networks

[*] Correspondence: gokhan.yetgin@dpu.edu.tr

and various evolutionary algorithms were used with both online and offline methods. In the literature, Wishart and Harley [5] experimented with a method that uses artificial neural networks for induction motor parameter estimation in current and speed control [6]. Linear techniques based on dynamic model and neuro-fuzzy methods are also proposed for the estimation of induction motor parameters [7,8]. In [9–11], the estimation of the stator resistance, transient inductance, and rotor resistance online were discussed. An interesting approach for tuning the rotor resistance was proposed in [12] based on model reference adaptive system schemes [13]. Since some of the above approaches require a derivative of the function, which is not always available or may be difficult to calculate, deterministic approaches often cannot find optimal solutions [1].

Recently, in solving induction motor parameter estimation problems, some new global optimization techniques, such as the evolutionary algorithm [14], genetic algorithm (GA) [15,16], differential evolution [17], particle swarm optimization (PSO) [18], ant colony optimization [19], harmony search (HS) [20] and big bang-big crunch [21], hybrid GA [22], and dynamic encoding algorithm for search [23], were proposed [1]. Moreover, the charged system search (CSS) is the most recent metaheuristic algorithm, which utilizes the Newtonian motion law in additional to electrical physics laws to direct the agents in order to recognize the optimum locations [24].

In this study, the CSS, differential evolution algorithm (DEA), PSO, and GA techniques were applied to estimate induction motor parameters. In the implementation of the techniques, 2 different induction motors, the squirrel-cage rotor and wound-rotor, are used. The 30-kW wound-rotor induction motor parameters are taken from [25] and the 37-kW squirrel-cage induction motor parameters are taken from [6].

In the literature, the CSS algorithm is generally used in civil engineering problems. There is no study where the CSS algorithm was applied for the estimation of induction motor equivalent circuit parameters based on torque values ($T_{st}, T_{\max}$, andT_n). As a preliminary work, in this study, the CSS algorithm was applied to estimate induction motor parameters.

2. Optimization techniques

2.1. Charged system search

Kaveh and Talatahari proposed the CSS algorithm [24,26]. The use of this algorithm is growing and its application is extending to various optimization problems [27–35]. A typical algorithm for the CSS is shown in Figure 1.

The CSS algorithm depends on Coulomb and Gauss laws and movement governing the motion laws of Newtonian mechanics. This algorithm can be considered as a multiagent approach, where each agent is a charged particle (CP). Each CP is assumed as a sphere with radius a and a proper charge density, and can be expressed as follows [30]:

$$q_i = \frac{fit(i) - fit_{worst}}{fit_{best} - fit_{worst}},\ i = 1, 2, 3, ..., N. \tag{1}$$

Here, *fit* $_{best}$ and *fit* $_{worst}$ are the best and worst fitness values of all of the particles, *fit(i)* is the fitness of agent i, and N is the total number of CPs. The initial positions of the CPs in search space are determined randomly and Eq. (2) is used for determination.

$$x_{i,j}^{(o)} = x_{i,\min} + rand_{ij}.(x_{i,\max} - x_{i,\min}),\ i = 1, 2, 3, ..., N \tag{2}$$

Here, $x_{i,j}^{(0)}$, determines the initial value of variable number i for CP number j, $x_{i,min}$ and $x_{i,max}$ are the minimum and maximum allowed values for variable number I, and $rand_{ij}$ is a randomly generated number

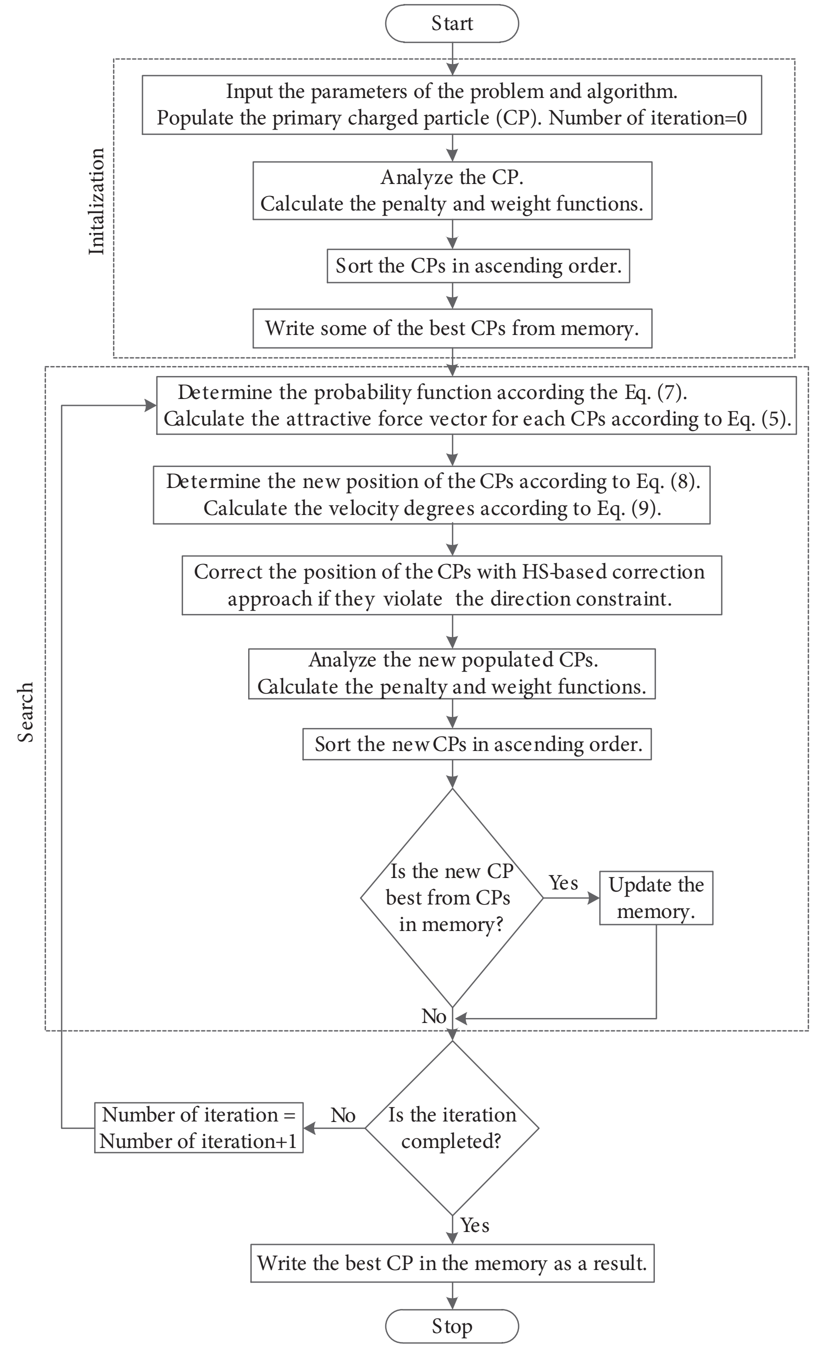
Start
Initalization
Input the parameters of the problem and algorithm.
Populate the primary charged particle (CP). Number of iteration=0
Analyze the CP.
Calculate the penalty and weight functions.
Sort the CPs in ascending order.
Write some of the best CPs from memory.
Search
Determine the probability function according the Eq. (7).
Calculate the attractive force vector for each CPs according to Eq. (5).
Determine the new position of the CPs according to Eq. (8).
Calculate the velocity degrees according to Eq. (9).
Correct the position of the CPs with HS-based correction approach if they violate the direction constraint.
Analyze the new populated CPs.
Calculate the penalty and weight functions.
Sort the new CPs in ascending order.
Is the new CP best from CPs in memory?
Yes
Update the memory.
No
Is the iteration completed?
No
Number of iteration = Number of iteration+1
Yes
Write the best CP in the memory as a result.
Stop

Figure 1. Flowchart of the CSS [27].

within the interval (0,1). The initial velocities of the CPs are taken as below:

$$v_{i,j}^{(o)} = 0,\ i = 1, 2, 3, ..., N. \tag{3}$$

Each CP applies a force on the other CPs according to Coulomb's law. The magnitude of this force is proportional with the distance between the CPs for the CP within the sphere, while it is inversely proportional with the square of the distance between the particles for a located CP outside of the sphere. These forces may come out as attracting or repelling and can be found with the ar_{ij} force parameter, defined as below:

$$ar_{ij} = \begin{cases} +1 & k_t < rand_{ij} \\ -1 & k_t > rand_{ij} \end{cases} . \tag{4}$$

The +1 value in Eq. (4) shows that the force is attracting and the –1 value shows that the force is repelling, and k_t is the parameter controlling the effect of the force type. Usually, the force coming out as attracting gathers the CPs in a certain area within the search area, while the repelling force tries to distribute the CPs. As a result, the force can be defined as below:

$$\mathbf{F}_j = \sum_{i,i\neq j} \left(\frac{q_i}{a^3} r_{ij}.i_1 + \frac{q_i}{r_{ij}^2}.i_2 \right).ar_{ij}.p_{ij}.\left(X_i - X_j\right) \quad \left\langle \begin{array}{l} j = 1,2,3,...,N \\ i_1 = 1,\ i_2 = 0 \Leftrightarrow r_{ij} < a \\ i_1 = 0,\ i_2 = 1 \Leftrightarrow r_{ij} \geq a \end{array} \right. . \tag{5}$$

Here, $\boldsymbol{F}_j$ is the force value acting on the jth CP and r_{ij} is the distance between 2 CPs, defined as follows:

$$r_{ij} = \frac{\|X_i - X_j\|}{\|(X_i - X_j)/2 - X_{best}\| + \varepsilon}. \tag{6}$$

Here, $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$ are the positions of CPs i and j, respectively; $\boldsymbol{X}_{best}$ is the position of the best current CP; and ε is a small positive number taken to prevent singularity. p_{ij} determines the moving possibility of each CP to the others, as below:

$$p_{ij} = \begin{cases} 1 & \frac{fit(i)-fit_{best}}{fit(j)-fit(i)} > rand \vee fit(i) > fit(j) \\ 0 & else \end{cases} . \tag{7}$$

As a result, the forces coming out and the motion laws determine the new CP positions. At this stage, each CP moves toward its new position under the effect of the forces and its previous velocity, as below:

$$X_{j,new} = rand_{j1}.k_a.\frac{\mathbf{F}_j}{m_j}.\Delta t^2 + rand_{j2}.k_v.\mathbf{V}_{j,old}.\Delta t + X_{j,old}, \tag{8}$$

$$\mathbf{V}_{j,new} = \frac{X_{j,new} - X_{j,old}}{\Delta t}. \tag{9}$$

Here, k_a is the acceleration coefficient, k_v is the velocity coefficient controlling the influence of the previous velocity, and $rand_{j1}$ and $rand_{j2}$ are 2 random numbers distributed to the sequence uniformly within the interval (0,1). If each CP moves out of the CP search space, its position is corrected by a handling approach based on HS. Moreover, the charged memory is used for recording the best results.

2.2. Differential evolution algorithm

The DEA is a heuristic optimization technique depending on the GA in the means of operation. It was developed by Storn and Price in 1995 [36]. Specifically, in problems where continuous data are in question, it gives efficient results. A new individual is obtained by putting chromosomes into the operators one by one (not operating depending on population). During this operation, mutation and crossover operators are used. If the convenience of the new individual is better than that of the old one, the old individual is conveyed to the next generation [37]. The flowchart of the algorithm is given in Figure 2 [38].

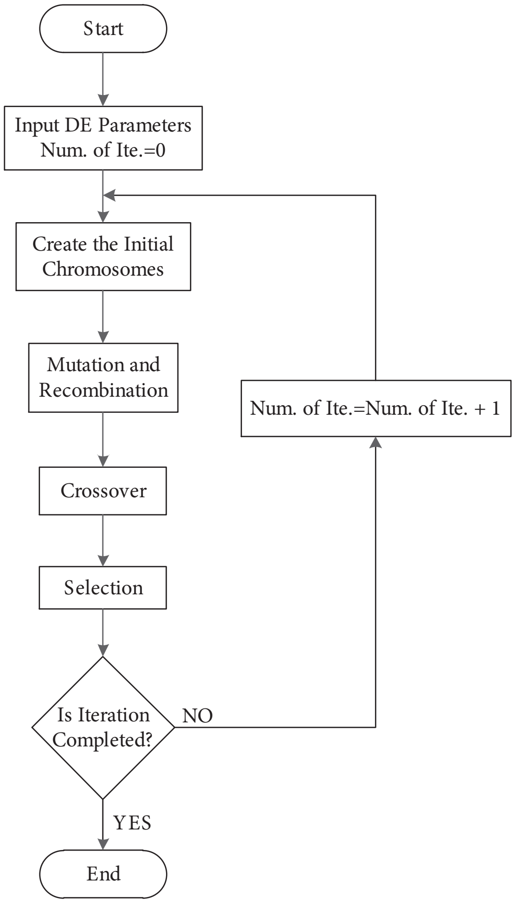

Figure 2. Flowchart of the DEA.

The parameters used in the DEA are population size *NP*, number of variables (number of genes) D, generation (1, 2, 3,. . . , $g_{\max}$)g, crossover rate *CR*, and scaling factor F [38].

The operation steps of the DEA are creation of the initial population, mutation, crossover, and selection. The conduction of these operations is explained below [39–45].

2.2.1. Creation of the initial population

In order to produce new chromosomes in the DEA, 3 chromosomes, with the exception of the corresponding chromosome, are needed. Therefore, the population size should be greater than 3 ($NP > 3$). The production of the initial population, consisting of *NP* pieces of chromosomes with D dimensions, is found using Eq. (10).

$$x_{j,i,g=0} = x_j^{(l)} + rand_j[0,1].(x_j^{(u)} - x_j^{(l)}) \tag{10}$$

Here, $x_{j,i,g}$ is the j parameter of the i chromosome in the g generation, and $(x_j^{(l)}, x_j^{(u)})$shows the lower and upper values of the variables.

2.2.2. Mutation

Mutation is making random changes to the chromosome genes. In the DEA, 3 chromosomes that are different from each other and the weighted difference chromosome are selected for mutation $(r_1, r_2,\ r_3)$. The difference of the first 2 is taken and is multiplied by F. In general, F is valued at between 0 and 2. The weighted difference chromosome and third chromosome are added.

$$n_{j,i,g+1} = x_{j,r_3,g} + F.(x_{j,r_1,g} - x_{j,r_2,g}) \tag{11}$$

Here, $n_{j,i,g+1}$ is the intermediate chromosome exposed to the *g + 1* mutation and crossing, and $r_{1,2,3} \in \{1, 2, 3, \ldots, NP\}\, r_1 \neq r_2 \neq r_3 \neq i$are the randomly chosen chromosomes that will be used for new chromosome generation.

2.2.3. Crossover

The different chromosomes produced by the mutation and the $x_{i,g}$ chromosome are used to produce a new chromosome $(u_{i,g+1})$. Genes for the trial are selected from different chromosomes with CR possibilities and from the corresponding chromosome with (1 – CR) possibility. The $j = j_{rand}$ condition is used for guaranteeing at least one gene to be taken from the recently produced chromosome. The randomly selected gene in the $j = j_{rand}$ point is selected from $n_{j,i,g+1}$ without taking the CR value into consideration.

$$x_{j,u,g+1} = \left\{ \begin{array}{l} x_{j,n,g+1} \;\; If \; rand[0,1] \leq CR \, or \, j = j_{rand} \\ x_{j,i,g} \quad otherwise \end{array} \right\} \tag{12}$$

2.2.4. Fitness function

Using the mutation and crossover, 3 chromosomes are used together with the target chromosome; a new (trial) chromosome is obtained. The chromosome that will be conveyed to the new generation $(g = g{+}1)$ is determined by looking at the suitability value. The fitness function of the target chromosome is already known. The objection function value of the problem is calculated as a suitability function.

2.2.5. Selection

The highly suitable chromosome is conveyed to the next generation. The cycle continues until $(g = g_{\max})$ and, when the cycle becomes $g_{\max}$, the current best individual is taken as the solution.

$$x_{i,g+1} = \left\{ \begin{array}{l} x_{u,g+1} \;\; If \;\; f(x_{u,g+1}) \leq f(x_{i,g+1}) \\ x_{i,g} \quad else \end{array} \right\} \tag{13}$$

2.2.6. Stopping criterion

The aim is continuously acquiring chromosomes with better suitability values and having the optimum value (or getting close). This cycle is continued until $g = g_{\max}$ and stopping the algorithm depends on the determined maximum iteration number.

The GA and PSO process algorithms, which are used for parameter estimation, were taken, respectively, from [46] and [47], and the parameter estimation of the induction motor was done using the formulas therein.

3. Problem formulation

The actual values of the starting torque T_{st} *(act)*, maximum torque $T_{\max}$ *(act)*, and nominal torque T_n *(act)* are used to estimate the stator resistance and leakage reactance (R_1, X_1), rotor resistance and leakage reactance $(R_2,\ X_2)$, and magnetizing reactance (X_m) parameters. The actual values are the literature values of the motors. The induction motor one-phase exact equivalent circuit model is shown in Figure 3.

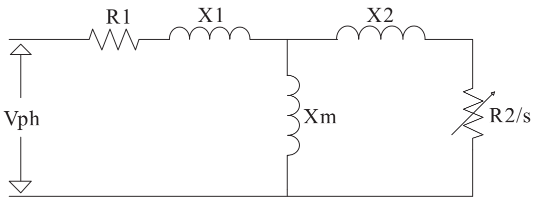

Figure 3. One-phase exact equivalent circuit model.

The torque functions [$T_n\,(cal)$,$T_{st}\,(cal)$, and $T_{\max}\,(cal)$] can be written as follows [48]:

$$T_n\,(cal) = \frac{KR_2}{s\left[\left(R_{th}+\frac{R_2}{s}\right)^2+X^2\right]}, \tag{14}$$

$$T_{st}\,(cal) = \frac{KR_2}{\left(R_{th}+R_2\right)^2+X^2}, \tag{15}$$

$$T_{\max}\,(cal) = \frac{K}{2\left[R_{th}+\sqrt{R_{th}^2+X^2}\right]}. \tag{16}$$

Here, $T_n\,(cal)$,$T_{st}\,(cal)$, and$T_{\max}\,(cal)$ stand for the nominal, starting, and maximum torques, respectively, with the following set of equations:

$$K = \frac{3V_{th}^2}{\omega_s}, V_{th} = \frac{X_m}{X_1+X_m}V_{ph}, R_{th} = \frac{X_m}{X_1+X_m}R_1, X_{th} = \frac{X_m}{X_1+X_m}X_1, X = X_2+X_{th}. \tag{17}$$

Here, V_{th}, R_{th}, and X_{th} are the Thevenin voltage, resistance, and reactance, respectively. K is the constant coefficient, ω_s is the angular velocity, s is the slip, and V_{ph} is the supply voltage. The fitness value of each of the motor torque equations is given in Eq. (18).

$$\begin{gathered} E_1 = \left|\frac{T_n(act)-T_n(cal)}{T_n(act)}\right| \\ E_2 = \left|\frac{T_{st}(act)-T_{st}(cal)}{T_{st}(act)}\right| \\ E_3 = \left|\frac{T_{\max}(act)-T_{\max}(cal)}{T_{\max}(act)}\right| \end{gathered} \tag{18}$$

Here, E_1, E_2, and E_3 show the error in the nominal torque, error in the starting torque, and error in the maximum torque, respectively. The total error (E_T) is given in Eq. (19).

$$E_T = |E_1+E_2+E_3| \tag{19}$$

The specifications of the motors used in the modeling are given in Table 1.

Table 1. Specifications of the motors.

Specifications	Motor 1	Motor 2
Nominal power (kW)	30	37
Nominal voltage (V)	460	460
Nominal frequency (Hz)	60	60
Number of poles	4	4
Nominal speed (rpm)	1740	1705
Cage type	Wound-rotor	Squirrel-cage

In all of the methods, at the beginning of the estimation process, the equivalent circuit parameter values are assigned randomly. Error values are checked at every iteration and the minimum error values are obtained by keeping the best parameter values. The equivalent circuit parameters of the 2 motors obtained with the 4 methods and actual values of these parameters are given in Table 2. The actual torque values and calculated torque values are shown in Table 3.

Table 2. Comparison of CSS, DEA, PSO, and GA equivalent circuit parameter results with the actual values for motors 1 and 2.

	CSS					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
R_1	0.25	0.2449278895	2.028844	0.087	0.082946564751	4.659120
R_2	0.2	0.1894845257	5.257737	0.228	0.221580835077	2.815423
X	1	0.9532414585	4.675854	0.604	0.606414293453	0.399717
X_m	30	9.1881489419	69.372836	13.08	13.730553467747	4.973650
	DEA					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
R_1	0.25	0.2486516245	0.539350	0.087	0.0835084377	4.013289
R_2	0.2	0.1952899383	2.355030	0.228	0.2241465675	1.690101
X	1	0.9866666553	1.333334	0.604	0.6070739183	0.508926
X_m	30	33.888837422	12.962791	13.08	13.0812247645	0.009363
	PSO					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
R_1	0.25	0.2451853615	1.9258554	0.087	0.0839569063	3.497808
R_2	0.2	0.1898968813	5.05155935	0.228	0.2264489983	0.680263
X	1	0.9570055972	4.29944028	0.604	0.6092982790	0.877198
X_m	30	10.9932328108	63.35589063	13.08	13.1422295778	0.475761
	GA					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
R_1	0.25	0.253165663877	1.26626555	0.087	0.0861628266	0.962268
R_2	0.2	0.202805246167	1.40262308	0.228	0.2375407538	4.184541
X	1	1.00624377900	0.62437790	0.604	0.6296539100	4.247336
X_m	30	14.749359845505	50.83546718	13.08	15.0626354814	15.15776

Table 3. Comparison of CSS, DEA, PSO, and GA torque results with the actual values for motors 1 and 2.

	CSS					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
T_{st}	163.11	163.1099999558	2.708356e-8	529.708	529.7080887509	1.675470e-5
$T_{\max}$	431.68	431.6800000193	4.486425e-9	773.987	773.9868719304	1.654673e-5
T_n	185.20	185.1999999592	2.198434e-8	234.55	234.5501880796	8.018743e-5
	DEA					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
T_{st}	163.11	163.1099999996	2.452332e-10	529.708	529.7079528896	8.893638e-6
$T_{\max}$	431.68	431.6800000121	2.803928e-11	773.987	773.9869326887	8.696692e-6
T_n	185.20	185.2000000001	8.153347e-11	234.55	234.5497933898	8.808789e-5
	PSO					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
T_{st}	163.11	163.1065040937	2.143281e-3	529.708	529.7043502484	6.890119e-4
$T_{\max}$	431.68	431.6801038641	2.406043e-5	773.987	773.9652578505	2.809110e-3
T_n	185.20	185.2026973711	1.456463e-3	234.55	234.5535738994	1.523726e-3
	GA					
	Motor 1			Motor 2		
Parameters	Actual values	Estimation values	Error (%)	Actual values	Estimation values	Error (%)
T_{st}	163.11	163.089203340828	12.75008e-3	529.708	529.7145521087	1.236928e-3
$T_{\max}$	431.68	431.690235169530	2.371008e-3	773.987	773.9774822166	1.229708e-3
T_n	185.20	185.191088011981	4.812088e-3	234.55	234.5654525646	6.588175e-3

When the results obtained for both motors are examined, it is seen that the error values of the equivalent circuit parameters are at acceptable levels. Here, compared with the other parameters, the error values of the X_m parameter are greater. However, the X_m value has no direct effect on the torque formulation. Thus, the error values of theX_m parameter can be ignored.

When examined in terms of the torque calculation, it is seen that the DEA obtains the best result. The torque values obtained by the CSS are close to those of the DEA. The biggest error values are obtained from the GA method.

The change of fitness values according to the iteration is shown in Figures 4 and 5 for motors 1 and 2, respectively.

When Figure 4 is examined, it is seen that the DEA method converges at the smallest iteration number. After the DEA, the CSS algorithm has the best convergence value for motor 1. It is seen that the PSO algorithm converges at the 125th iteration. However, it is seen that the GA is not able to reach the convergence values.

When Figure 5 is examined, it is seen that the DEA again has the best convergence. It is seen that the CSS algorithm similarly converges to the motor 2 value at the 55th iteration, and the PSO algorithm converges at the 130th iteration. It is also seen that the GA does not converge.

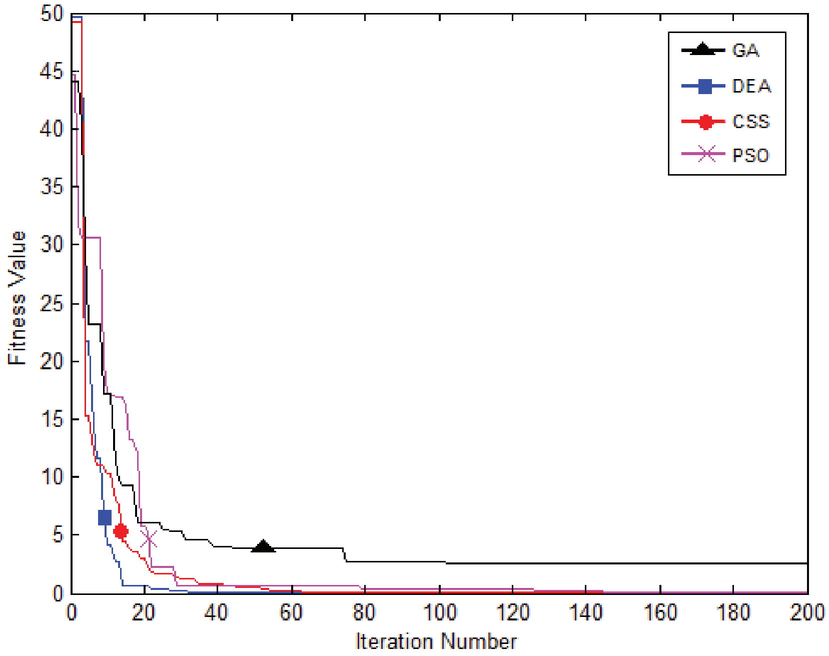

Figure 4. Iteration number versus the fitness value for motor 1.

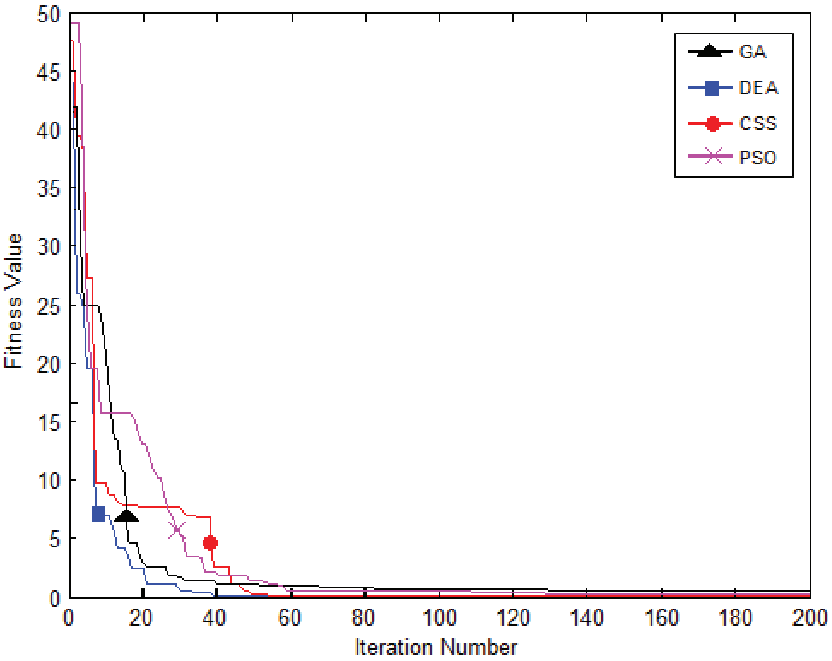

Figure 5. Iteration number versus the fitness value for motor 2.

The slip-torque curves drawn depending on the parameters found by the actual torque values and estimation values are shown in Figures 6 and 7. When examined in terms of the minimum error values, the 4 methods are seen to have caught the actual values with very small differences.

The total error values that occurred in the torque values for motor 1 for each method are shown in Figure 8.

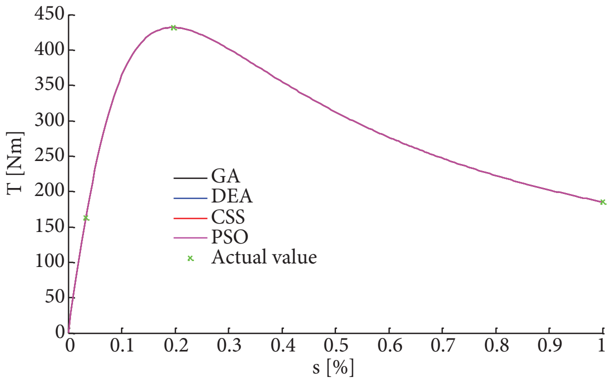

Figure 6. Slip versus torque for motor 1.

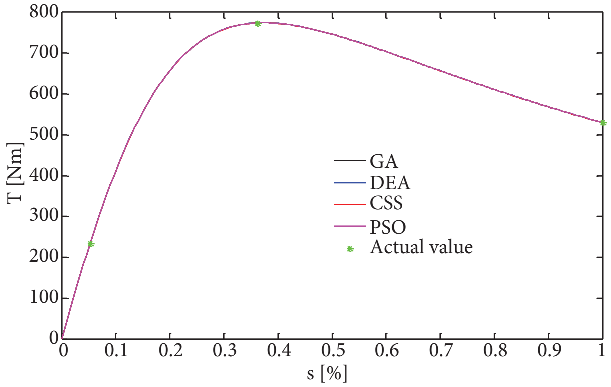

Figure 7. Slip versus the torque for motor 2.

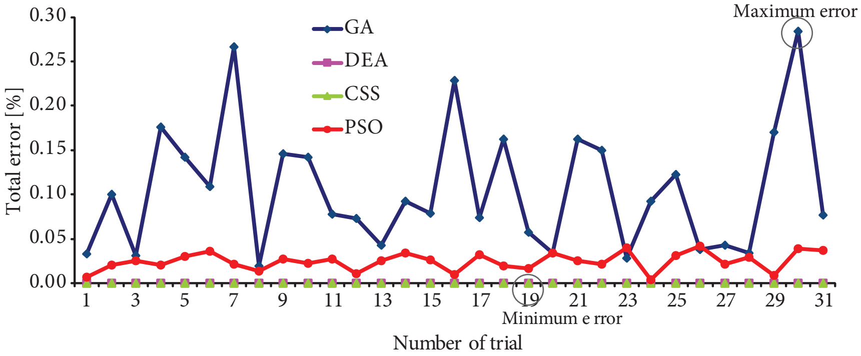

Figure 8. Distribution of the total error of the 4 methods for motor 1 (for torque values).

When the total error values given for motor 1 are examined, it is seen that the values obtained by the GA as a result of 31 operations are bigger than the values obtained by the other 3 methods. It is seen that the maximum error value is obtained by the GA at the 30th operation, and the minimum error value is obtained at the 8th operation. It is seen that, after the GA, the biggest error values are obtained by the PSO algorithm. However, the lowest error values are obtained in the parameter estimation done by the DEA and CSS algorithm. In the operations done by the 4 methods, the smallest error value is obtained by the DEA at its 19th operation. However, it is seen that the examined CSS algorithm has values close to those of the DEA, which gives the smallest error values.

The total error values that occurred in the torque values for motor 2 for each method are shown in Figure 9.

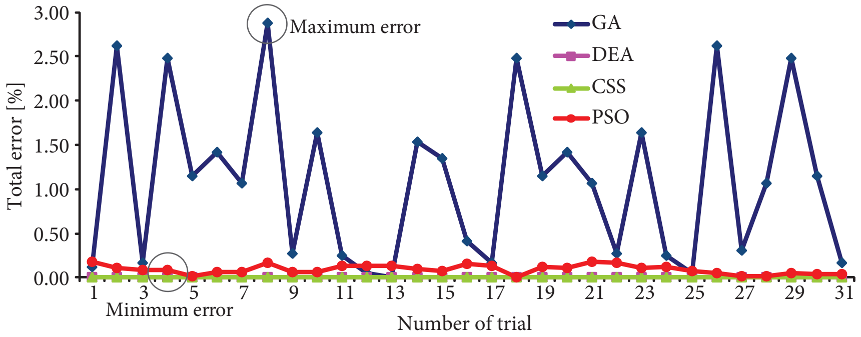

Figure 9. Distribution of the total error of the 4 methods for motor 2 (for torque values).

When all of the error values given for motor 2 are examined, it is seen that the values are similar to the distribution of total error obtained for motor 1. While the minimum error value is obtained from the DEA at its 4th operation, the maximum error value is obtained by the GA at its 8th operation. The error value of the CSS algorithm is close to that of the DEA.

The CPU times obtained by the 31 operations of the 4 methods are given in Figures 10 and 11. It is seen that the GA has the longest solution time and the PSO has the shortest solution time. The solution times of the 4 methods are seen to be less than 1 s for both motors.

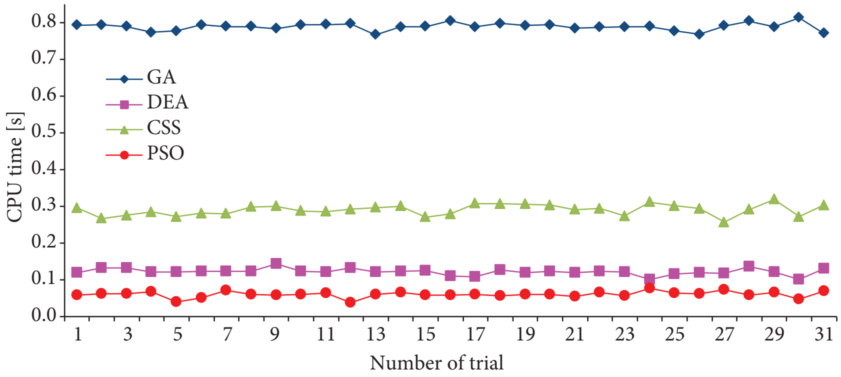

Figure 10. Distribution of the CPU time of the 4 methods for motor 1.

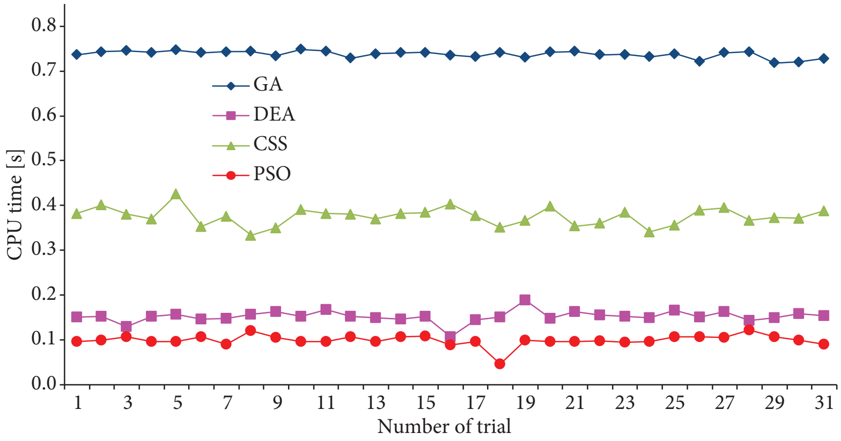

Figure 11. Distribution of the CPU time of the 4 methods for motor 2.

When Table 4 is examined, it is seen that the maximum total error value is obtained from the GA and the total minimum error value is obtained from the DEA. The CSS algorithm gives the best results after the DEA. When the standard deviation values are examined, the DEA and CSS algorithm have the lowest standard deviation at 31 operations.

Table 4. Comparison of the CSS performance with other the methods for motor 1 (for torque values).

Methods	Maximum error	Average error	Minimum error	Standard deviation
CSS	0.000017208796	0.000005448508	0.000000053554	4.85183e-06
DEA	0.000000193725	0.000000082610	0.000000003131	6.13198e-08
PSO	0.042305520373	0.024503987984	0.003623805767	0.010199809
GA	0.284213252677	0.106073346508	0.019933179330	0.070428281

The equivalent circuit parameters and average percent error values obtained by the DEA and GA in [6] are given comparatively in Table 5, with the values found by the CSS algorithm in this study.

Table 5. Comparison of the equivalent circuit parameters (DEA, GA, and CSS).

		Reference values [6]				Present study's values	
		DEA		GA		CSS	
Parameters	Actual values	Estimation values	Error (%)	Estimation values	Error (%)	Estimation values	Error (%)
R_1	0.087	0.087	0	0.084	3.448	0.082…	4.659…
R_2	0.228	0.238	4.385	0.239	4.824	0.221…	2.815…
X	0.604	0.631	4.470	0.633	4.801	0.606…	0.399…
X_m	13.08	13.291	1.163	13.304	1.712	13.730…	4.973…
Average error			2.617		3.696		3.211

The error value found by the CSS is seen to converge to the average error value found by the DEA method used in [6]; however, it is seen to converge with a better result than that found by the GA.

When the torque values in Table 6 are considered, the error values found by the CSS are seen to give better results than those found by the DEA and GA.

Table 6. Comparison of the torque values (DEA, GA, and CSS).

		Reference values [6]				Present study's values	
		DEA		GA		CSS	
Parameters	Actual values	Estimation values	Error (%)	Estimation values	Error (%)	Estimation values	Error (%)
T_{st}	529.708	531.659	0.368	530.635	0.175	529.708	1.675e-5
$T_{\max}$	773.987	776.352	0.305	777.149	0.408	773.986	1.654e-5
T_n	234.55	234.863	0.133	235.076	0.224	234.550	8.018e-5
Average error			0.268		0.269		3.782e-5

4. Conclusion

In this study, the CSS algorithm for estimation of the equivalent circuit parameters of an induction motor was applied for the first time in the literature. The obtained results were compared with the DEA, PSO, and GA for 2 different motors. The known equivalent circuit parameters and slip-torque characteristics of wound-rotor motor were obtained from [25] and the required values were taken from these characteristics. For the squirrel-cage motor, the equivalent circuit parameters were obtained from [6]. The reason for using the literature data is that the catalog values given by the manufacturer cannot model the motor properly. It is not possible to obtain convergence from the motor catalog values of various manufacturers.

For motor 1, the CSS algorithm and DEA converge faster than the other algorithms. For motor 2, the CSS algorithm catches the optimum point at 55 iterations and the DEA catches the optimum point at 40 iterations. As seen from Figures 6 and 7, the estimated equivalent circuit parameters exactly model the slip-torque characteristics of both motors. From the obtained results, it is observed that if manufacturers give the exact values for motors, the CSS algorithm can estimate the equivalent circuit parameters properly.

This study shows hopeful results that the CSS algorithm can be used to estimate the parameters of other types of electrical machines as well.

References

[1] V.P. Sakthivel, R. Bhuvaneswari, S. Subramanian, "Multi-objective parameter estimation of induction motor using particle swarm optimization", Engineering Applications of Artificial Intelligence, Vol. 23, pp. 302–312, 2010.

[2] S. Kumar, J. Prakash, P. Kanagasabapathy, "A critical evaluation and experimental verification of extended Kalman filter, unscented Kalman filter and neural state filter for state estimation of three phase induction motor", Applied Soft Computing, Vol. 11, pp. 3199–3208, 2011.

[3] Y. Koubaa, "Application of least-squares techniques for induction motor parameters estimation", Mathematical and Computer Modelling of Dynamical Systems, Vol. 12, pp. 363–375, 2006.

[4] J. Zheng, Y. Wang, X. Qin, X. Zhang, "An offline parameter identification method of induction motor", Proceedings of the 7th World Congress on Intelligent Control and Automation, pp. 8898–8901, 2008.

[5] M.T. Wishart, R.G. Harley, "Identification and control of induction machines using artificial neural networks", IEEE Transactions on Industry Applications, Vol. 31, pp. 612–619, 1995.

[6] M. Arslan, M. Çunkaş, T. Sağ, "Determination of induction motor parameters with differential evolution algorithm", Neural Computing and Applications, Vol. 21, pp. 1995–2004, 2012.

[7] L.A. De Souza Ribeiro, C.B. Jacobina, A.M.N. Lima, "Linear parameter estimation for induction machines considering the operating conditions", IEEE Transactions on Power Electronics, Vol. 14, pp. 62–73, 1999.

[8] L. Rajaji, C. Kumar, "Adaptive neuro-fuzzy inference systems into squirrel cage induction motor drive: modeling, control and estimation", 5th International Conference on Electrical and Computer Engineering, pp. 162–169, 2008.

[9] M.A. Alwan, "Online tuning of stator resistance in DTC drive based on wavenet theory", Journal of Basrah Researches (Sciences), Vol. 37, pp. 27–41, 2011.

[10] M.H. Shin, D.S. Hyun, "Online identification of stator transient inductance in rotor-flux-oriented induction motor drive", IEEE Transactions on Industrial Electronics, Vol. 54, pp. 2018–2023, 2007.

[11] S.H. Jeon, K.K. Oh, J.Y. Choi, "Flux observer with online tuning of stator and rotor resistances for induction motors", IEEE Transactions on Industrial Electronics, Vol. 49, pp. 653–664, 2002.

[12] Y. Koubaa, M. Boussak, "Adaptive rotor resistance identification for indirect stator flux oriented induction motor drive", Proceedings of the 2nd International Conference on Signals, Systems Decision and Information Technology, 2003.

[13] B. Subudhi, D. Jena, "Differential evolution computation applied to parameter estimation of induction motor", Archives of Control Sciences, Vol. 19, pp. 5–26, 2009.

[14] T.O. Kowalska, J. Lis, K. Szabat, "Identification of the induction motor parameters at standstill using soft computing methods", COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, Vol. 25, pp. 181–194, 2006.

[15] S. Jangjit, P. Laohachai, "Parameter estimation of three-phase induction motor by using genetic algorithm", Journal of Electrical Engineering & Technology, Vol. 4, pp. 360–364, 2009.

[16] K. Kampisios, P. Zanchetta, C. Gerada, 13th International A. Trentin, O. Jasim, "Induction motor parameters identification using genetic algorithms for varying flux levels", Power Electronics and Motion Control Conference, pp. 887–892, 2008.

[17] R. Giri, A. Chowdhury, A. Ghosh, B.K. Panigrahi, S. Das, "Offline parameter estimation of induction motor using a meta heuristic algorithm", Lecture Notes in Computer Science, Vol. 6466, pp. 523–530, 2010.

[18] V.P. Sakthivel, R. Bhuvaneswari, S. Subramanian, "An improved particle swarm optimization for induction motor parameter determination", International Journal of Computer Applications, Vol. 1, pp. 62–67, 2010.

[19] Z. Chen, Y. Zhong, J. Li, "Parameter identification of induction motors using ant colony optimization", IEEE World Congress on Computational Intelligence, pp. 1611–1616, 2008.

[20] Z.W. Geem, J.H. Kim, G.V. Loganathan, "A new heuristic optimization algorithm; harmony search", Simulation, Vol. 76, pp. 60–68, 2001.

[21] O.K. Erol, I. Eksin, "New optimization method: big bang-big crunch", Advances in Engineering Software, Vol. 37, pp. 106–111, 2006.

[22] K. Sundareswaran, H.N. Shyam, S. Palani, J. James, "Induction motor parameter estimation using hybrid genetic algorithm", IEEE Region 10 and the 3rd International Conference on Industrial and Information Systems, pp. 1–6, 2008.

[23] J.W. Kim, S.W. Kim, "Parameter identification of induction motors using dynamic encoding algorithm for searches (DEAS)", IEEE Transactions on Energy Conversion, Vol. 20, pp. 16–24, 2005.

[24] A. Kaveh, S. Talatahari, "An enhanced charged system search for configuration optimization using the concept of fields of forces", Structural and Multidisciplinary Optimization, Vol. 43, pp. 339–351, 2011.

[25] P.C. Sen, Principles of Electric Machines and Power Electronics, New York, NY, Wiley & Sons, 1997.

[26] A. Kaveh, S. Talatahari, "A novel heuristic optimization method: charged system search", Acta Mechanica, Vol. 213, pp. 267–289, 2010.

[27] A. Kaveh, S. Talatahari, "Optimal design of skeletal structures via the charged system search algorithm", Structural and Multidisciplinary Optimization, Vol. 41, pp. 893–911, 2010.

[28] A. Kaveh, S. Talatahari, "Charged system search for optimum grillage systems design using the LRFD-AISC code", Journal of Constructional Steel Research, Vol. 66, pp. 767–771, 2010.

[29] A. Kaveh, S. Talatahari, "Geometry and topology optimization of geodesic domes using charged system search", Structural and Multidisciplinary Optimization, Vol. 43, pp. 215–229, 2011.

[30] S. Talatahari, A. Kaveh, R. Sheikholeslami, "An efficient charged system search using chaos for global optimization problems", International Journal of Optimization in Civil Engineering, Vol. 2, pp. 305–325, 2011.

[31] A. Kaveh, S. Talatahari, "A charged system search with a fly to boundary method for discrete optimum design of truss structures", Asian Journal of Civil Engineering, Vol. 11, pp. 277–293, 2010.

[32] A. Kaveh, S. Talatahari, "Optimization of large-scale truss structures using modified charged system search", International Journal of Optimization in Civil Engineering, Vol. 1, pp. 15–28, 2011.

[33] A. Kaveh, S. Talatahari, "Charged system search for optimal design of frame structures", Applied Soft Computing, Vol. 12, pp. 382–393, 2012.

[34] A. Kaveh, K. Laknejadi, "A novel hybrid charge system search and particle swarm optimization method for multi-objective optimization", Expert Systems with Applications, Vol. 38, pp. 15475–15488, 2011.

[35] A. Kaveh, S. Talatahari, "Hybrid charged system search and particle swarm optimization for engineering design problems", International Journal for Computer-Aided Engineering and Software, Vol. 28, pp. 423–440, 2011.

[36] R. Storn, K. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces", Technical Report TR-95-012, International Computer Science Institute, Berkeley, CA, USA, 1995.

[37] T. Keskintürk, "Differential evolution algorithm", İstanbul Commerce University Journal of Science, Vol. 9, pp. 85–99, 2006.

[38] S. Özyön, C. Yaşar, H. Temurtaş, "Differential evolution algorithm approach to nonconvex economic power dispatch problems with valve point effect", 6th International Advanced Technologies Symposium, Electrical & Electronics Technologies Papers, pp. 181–186, 2011.

[39] A.S. Uyar, B. Türkay, "Evolutionary algorithms for the unit commitment problem", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 16, pp. 239–255, 2008.

[40] T. Keskintürk, "Differential evolution algorithm", YA/EM–Operational Research/Industrial Engineering–XXVI, National Congress, pp. 214–217, 2006.

[41] R. Thangaraj, M. Pant, C.T. Raj, A.K. Nagar, "In situ efficiency determination of induction motor: a comparative study of evolutionary techniques", Applied Artificial Intelligence, Vol. 25, pp. 116–140, 2011.

[42] M. Ali, M. Pant, A. Abraham, V. Snasel, "Modified differential evolution algorithm for parameter estimation in mathematical models", IEEE International Conference on Systems Man and Cybernetics, pp 2767–2772, 2010.

[43] C. Yaşar, H. Temurtaş, S. Özyön, "Differential evolution algorithm applied to environmental economic power dispatch problems consisting of thermal units", 6th National Electrical & Electronics and Computer Engineering Symposium, pp. 108–112, 2010.

[44] T. Keskintürk, "The traveling salesman problem solution by using differential evolution algorithm", 6th National Symposium on Production Research, pp. 651–657, 2006.

[45] D. Zoua, H. Liu, L. Gao, S. Li, "A novel modified differential evolution algorithm for constrained optimization problems", Computers and Mathematics with Applications, Vol. 61, pp. 1608–1623, 2011.

[46] C. Yaşar, S. Özyön, "Solution to scalarized environmental economic power dispatch problem by using genetic algorithm", Electrical Power and Energy Systems, Vol. 38, pp. 54–62, 2012.

[47] S. Özyön, C. Yaşar, H. Temurtaş, "Particle swarm optimization algorithm for the solution of nonconvex economic dispatch problem with valve point effect", 7th International Conference on Electrical and Electronics Engineering, pp. 101–105, 2011.

[48] M. Ojaghi, M. Mardani, "Parameter estimation of induction motor using shuffled frog leaping and imperialistic competitive algorithms", 26th International Power System Conference, pp. 1–9, 2011.