COMPARATIVE STUDY OF PHYSICS-INSPIRED META-HEURISTIC ALGORITHMS FOR THE SOLAR CELL PARAMETER IDENTIFICATION PROBLEM

Alkın Yurtkuran, İlker Küçükoğlu
Uludag University, Department of Industrial Engineering, Bursa, Turkey
Corresponding author: İlker Küçükoğlu, e-mail: ikucukoglu@uludag.edu.tr

REFERENCE NO

PCEL-01

Keywords:
Solar cell
Parameter identification
Optimization
Physics-inspired meta-heuristics

ABSTRACT

In recent years, clean and renewable energy generation has drawn significant attention. Among different renewable energy sources, solar energy is one of the most promising due to readily availability, low maintenance cost and zero pollution. In order to optimize the performance of solar systems, parameters of solar photovoltaic (PV) cells should be identified accurately. Parameter identification refers to the process of extracting the electrical parameters of the PV cells from measured Current vs. Voltage $(I-V)$ curves with minimum prediction error, which is called as solar cell parameter identification problem (SCPIP). This paper introduces a comparative analysis of four different physics-inspired meta-heuristics algorithms on SCPIP. In this context, Electromagnetic Field Optimization algorithm, Electromagnetism-like Algorithm, Gravitational Search Algorithm, and Weighted Superposition Attraction algorithm are taken into account. In computational studies, tests with each algorithm are carried out on a well-known benchmark data set. Results of the computational studies reveal that Electromagnetic Field Optimization algorithm outperforms other three competitor algorithms.

1. INTRODUCTION

Renewable energy has experienced a tremendous increase in recent decades because of the depletion of conventional sources like oil, coal or natural gas. Among various kinds of renewable energy sources, solar energy is the most important source due to its properties such as environmental-friendly, unlimited capacity and wide-scale availability, little maintenance, noise-free and, easy installation [1]. Solar energy is converted into electrical energy through photovoltaic (PV) systems such as solar cell. Solar cell modelling primarily involves the formulation of the non-linear relationship between current and voltage using the $I-V$ curve [2]. Within wide variety of models, two solar cell models widely used in real world: the single and double diode models. The single diode model has five unknown parameters and the double diode model has seven unknown parameters. Accurate determination of those parameters, which are usually not provided by manufacturers, is vital for solar cell performance optimization. The SCPIP covers the identification of the optimal parameter values for a given $I-V$ curve dataset with a minimum prediction error.

To solve the SCPIP, there exist several solution approaches in the literature, which are mainly divided into two groups: deterministic and heuristic solution approaches. Regarding the deterministic approaches, a number of methods are employed by the researchers, such as, nonlinear least-squares based on the Newton model [3], iterative curve fitting [4], Lambert W-function [5], $J-V$ model [6], etc. However, these deterministic solution approaches are not efficient to solve SCPIP since they need continuity, convexity and differentiability conditions for being applicable and involve heavy computations [1, 7]. To cope with the complexity of the SCPIP, heuristic methods are used as an alternative to deterministic solution approaches.

Regarding the popular meta-heuristic algorithms, such as, simulated annealing algorithm [8], genetic algorithm [9, 10], particle swarm optimization algorithm [11], differential evolution algorithm [12-14], artificial bee colony algorithm [15], are widely used for the SCPIP. In addition to these well-known heuristic algorithms, there exist several papers in the literature which consider more recent approaches, such as, bacterial foraging algorithm [16, 17], teaching learning-based optimization algorithm [18-20], biogeography-based optimization algorithm [21], bird mating optimizer approach [22], artificial immune system [23], cat swarm optimization algorithm [24], moth-flame optimization algorithm [25], JAYA optimization algorithm [2], chaotic whale optimization algorithm [26].

This study presents a comparative performance assessment of well-known physics-inspired meta-heuristic algorithms i.e. Electromagnetic Field Optimization (EFO), Electromagnetism-like Algorithm (EMA), Gravitational Search Algorithm (GSA), and Weighted Superposition Attraction algorithm (WSA) on single diode SCPIP. In order to compare the algorithms objectively, identical test environments, population sizes and, stopping conditions were employed. Furthermore, the control parameters of the algorithms are set to their original values given in their corresponding papers. Computational results show that the EFO achieves a superior performance to other competitor algorithms.

The rest of the paper is organized as follows. Section 2 describes the SCPIP considering single diode model. The considered physics-inspired meta-heuristic algorithms are presented in Section 3. Computational results are given in Section 4. Finally, conclusions with future research perspectives are given in Section 5.

2. PROBLEM DEFINITION

In order to describe the $I-V$ characteristics of the solar cells, there exist several models in the literature. In this study, only the single diode model is taken into account since this model is much more common compared to the double diode model [1].

Fig. 1 represents the equivalent circuit for the single diode model, where V_t is the terminal voltage, R_s is the series resistance, R_{sh} is the shunt resistance, I_t is the terminal current, I_{ph} is the photo-generated current, $I_{sd}$is the diode current, and I_{sh} is the shunt resistor current. By using the Shockley equation for the diodes currents, the single diode model can be formulated as shown in Eq. 1 [15, 20, 26]. According to the Shockley equation, q is the magnitude of charge on an electron ($1.60217646 \times 10^{-19}$ coulombs), k is the Boltzmann constant ($1.3806503 \times 10^{-23}\, J/K$), and T is the cell temperature in Kelvin [15].

$$I_t = I_{ph} - I_{sd}\left[exp\left(\frac{q(V_t + R_s.I_t)}{n.k.T}\right) - 1\right] - \left(\frac{V_t + R_s.I_t}{R_{sh}}\right) \quad (1)$$

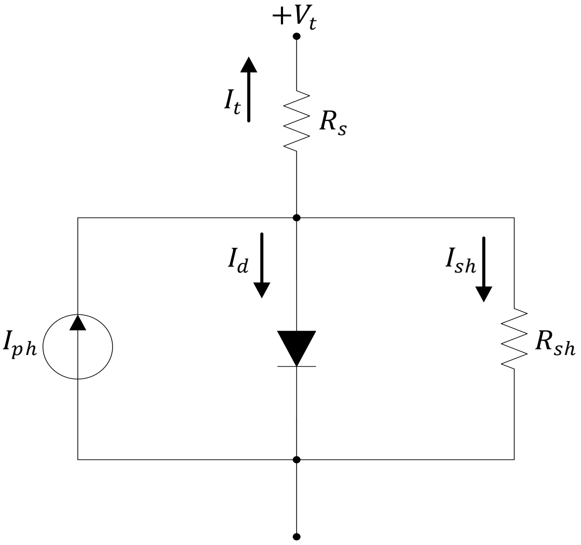

Fig. 1. Single diode model of solar cells [7, 20]

Regarding the single diode model described above, the SCPIP can be defined as identifying the parameters of Eq. 1 within their lower and upper bounds. The aim of the problem is to estimate the best parameter values for the single diode model that produce an accurate approximation between the $I-V$ measurements from the physical experiments and values from the mathematical model. Hence, the Eq. 1 can be rewritten as an error function as shown in Eq. 2.

$$f(V_t, I_t, R_s, R_{sh}, I_{ph}, I_{sd}, n) = I_t - I_{ph} + I_{sd}\left[exp\left(\frac{q(V_t + R_s.I_t)}{n.k.T}\right) - 1\right]$$

$$+\left(\frac{V_t + R_s.I_t}{R_{sh}}\right) \quad (2)$$

In the sense of optimization problem for accurate estimation, the error function for the SCPIP can be transformed to the Eq. 3 by using a decision variable vector $\boldsymbol{X}$, where Table 1 shows the descriptions of the decision variables and their lower and upper bounds.

$$f(V_t, I_t, \boldsymbol{X}) = I_t - x_3 + x_4\left[exp\left(\frac{q(V_t + x_1.I_t)}{x_5.k.T}\right) - 1\right] + \left(\frac{V_t + x_1.I_t}{x_2}\right) \quad (3)$$

Table 1. Lower and upper bounds of the solar cell parameters

Parameters of Single Diode Model	Decision Variable Vector $\boldsymbol{X}$	Lover Bound	Upper Bound
$R_s(\Omega)$	x_1	0	0.5
$R_{sh}(\Omega)$	x_2	0	100
$I_{ph}(A)$	x_3	0	1
$I_{sd}(\mu A)$	x_4	0	1
n	x_5	1	2

For a number of experimental data for a single diodes, the mathematical model formulation of the SCPIP can be defined as follows:

$$Min\, Z = RMSE(\boldsymbol{X}) \quad (4)$$

Subject to

$$0 \leq x_1 \leq 0.5 \quad (5)$$

$$0 \leq x_2 \leq 100 \quad (6)$$

$$0 \leq x_3 \leq 1 \quad (7)$$

$$0 \leq x_4 \leq 1 \quad (8)$$

$$1 \leq x_5 \leq 2 \quad (9)$$

$$x_1, x_2, x_3, x_4, x_5 \geq 0 \quad (10)$$

The objective function (4) aims to minimize RMSE (root mean square error) of the experimental, which is determined by using the Eq. 11 for a number of experiments, M. The constraints (5)-(9) describe the boundaries of the decision variables. Finally, the decision variables are described in constraint (10).

$$RMSE(\boldsymbol{X}) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(f_i(V_t, I_t, \boldsymbol{X})\right)^2} \quad (11)$$

3. PHYSICS-INSPIRED META-HEURISTIC ALGORITHMS

This section covers the basics of the physics-inspired meta-heuristic algorithms, which are investigated.

3.1. Electromagnetic Field Optimization Algorithm

The EFO is a population based meta-heuristic algorithm inspired by the electromagnets behaviour [27]. In EFO, a solution is represented by electromagnetic particle (EMP) made of electromagnets and the number of electromagnets is equal to the number of variables in the optimization problem. Different from permanent magnets, each electromagnet in EMP has same single polarity (positive or negative) and each electromagnet can apply a force of attraction or repulsion among other neighbour electromagnets. The main steps of the EFA can be described as follows:

Step 1. Generate an initial population of N electromagnetic particles, x_j^i, $i = 1 \ldots N, j = 1 \ldots D$ where x is the electromagnetic particle, N is the population size and D is the problem size.

Step 2. Calculate the objective value of each electromagnetic particle i $f(x^i)$ and sort the population according to the objective values descending.

Step 3. Divide the population into three groups. The first group is called the positive field, which contains better EMP's having positive polarity. The second group includes the worst EMP's with negative polarity. The third group called neutral field having the EMP's in between positive and negative fields.

Step 4. Select one EMP from each group randomly.

Step 5. The j^{th} index of the new EMP is copied directly from positive field EMP, if $rand(0,1) < ps\_rate$ where $rand(0,1)$ is a random number between 0 and 1, ps_rate is a control parameter. Else, the j^{th} index of the new particle is generated by using Eq. 12.

$$x_j^{new} = x_i^n + (\rho * r)\left(x_j^p - x_j^n\right) - r * \left(x_j^k - x_j^n\right) \quad (12)$$

where p, k and n are the selected positive, neutral and negative polarity EMP's respectively. r is a random number in the interval of $[0, 1]$ and ρ is the golden ratio. Step 5 is repeated for each index of the new particle.

Step 6. If $r\_rate < rand(0,1)$ then one electromagnet of the new EMP is replaced by a new randomly generated one. Here, r_rate is a control parameter.

Step 7. If the new EMP is better than the worst EMP in the current population, the new particle substitutes it and the population is sorted again. Steps 3-7 are repeated until a stopping condition is met.

3.2. Electromagnetism-Like Algorithm

The EMA is a population-based meta-heuristic algorithm proposed by [28] and simulates the electromagnetism theory, in which charged particles exert an attraction or repulsion forces on each other. In EMA, a solution is associated with a charged particle in a multi-dimensional space, x_{ij}, $i = 1, \ldots, N$ and $j = 1, \ldots, D$, where N and D are the population size and problem dimension respectively. The charge of each particle x_{i} depends on the quality of the fitness value $f(x_i)$. In EMA, each particle exerts a repulsion or attraction force on other population members according to the charges they carry. The results force F^i is determined by the sum of all forces on particle i and is moved in the direction of F^i. The idea of EMA is that, better particles will attract and worse particle will repel. The main steps of the algorithm are as follows:

Step 1. Generate an initial population of N particles, x_j^i, $i = 1 \ldots N, j = 1 \ldots D$ where x is the particle, N is the population size and D is the problem size.

Step 2. Calculate the fitness of the population and the charges of each particle, q^i by using Eq. 13.

$$q^i = exp\left(-D \frac{f\left(x^i\right) - f(x^{best})}{\sum_{k=1}^{N} f(x^k) - f(x^{best})}\right) \quad (13)$$

where x^{best} denotes the best particle in the current population.

Step 3. Calculate the resulting forces on particles for a minimization problem by using the Eq. 14 if $f(x^k) < fx^i)$, and Eq. 15 otherwise.

$$F^i = \sum_{k=1, k \neq i}^{N} x^k - x^i \frac{q^k q^i}{||x^k - x^i||} \quad (14)$$

$$F^i = \sum_{k=1, k \neq i}^{N} x^i - x^k \frac{q^k q^i}{||x^k - x^i||} \quad (15)$$

Step 4. After calculating the total force, re-position the particle as Eq. 16.

$$x^i = x^i + rand(0,1) * \frac{F^i}{||F^i||} \quad (16)$$

Step 5. Repeat Steps 2-5 until a pre-defined stopping condition is met.

3.3. Gravitational Search Algorithm

The GSA is a physics inspired meta-heuristic algorithm, which is based on the Newtonian laws of gravitation and motion [29]. In GSA each solution is represented by particles and has properties as position, active gravitational mass, passive gravitational mass and inertial mass where the position corresponds to the solution. Moreover, the gravitational masses and inertial mass is proportional to the fitness value. The key idea behind the GSA is that each particle attracts every other particle and heavier or better particles attracts more. The basic steps of the GSA are as follows:

Step 1. Generate an initial population randomly. The j^{th} index of particle i is denoted by x_i^j, $i = 1, \dots, N, j = 1, \dots, D$ where N is the population size and D is the number of variables in the optimization problem.

Step 2. Determine the fitness of the population.

Step 3. Determine the force that acts on the j^{th} index of the particle i by using Eq. 17.

$$F_i^j = \sum_{j=1, j\neq i} rand(0,1) * F_{ik}^j \qquad (17)$$

where $rand(0,1)$ is a random number in the interval $[0, 1]$, and the force exerted by particle k on i for thej^{th} index, the determination of F_{ik}^j is given in Eq. 18:

$$F_{ik}^j = G * \frac{M_i * M_j}{r_{ik} + \epsilon}\left(x_k^j - x_i^j\right) \qquad (18)$$

where ϵ is a constant, M_i and M_j are the masses of particle i and j respectively. G is the gravitational constant. The mass of the particle i, M_i is calculated by using Eq. 19 where $m_i = \frac{f(x_i) - f(x_{best})}{f(x_{best}) - f(x_{worst})}$.

$$M_i = {}^{m_i}\!/\!{}_{\sum_{k=1}^{N} m_k} \qquad (19)$$

Step 4. Determine the acceleration of the particle in j^{th} dimension as: $a_i^j = \frac{F_i^j}{M_i}$

Step 5. Re-position the particle i by mean of acceleration, as: $v_i^j = rand(0,1) * v_i^j + a_i^j$ and $x_i^j = x_i^j + v_i^j$.

Step 6. Repeat the step 3-5 until any stopping criteria is met.

3.4. Weighted Superposition Attraction Algorithm

Baykasoğlu and Akpinar [30] proposed the WSA, which adopts the superposition principle in combination with the attracted movements of agents. In WSA each solution is represented by an agent x_i where $i = 1, \dots, N$ and N is the population size. WSA determines searching directions of the agents by realizing combined superposition with attraction motions of agents through a neighbour generation mechanism [31]. The main steps of the WSA are given as below:

Step 1. Generate an initial population randomly. The j^{th}h index of particle i is denoted by x_{ij}, $i = 1, \dots, N, j = 1, \dots, D$ where N is the population size and D is the number of variables in the optimization problem.

Step 2. Determine the fitness of the population.

Step 3. Sort the population according to the fitness values.

Step 4. Calculate the weight of each agent using the rank i as $w_i = i^{-\tau}$ where τ is a control parameter of the algorithm.

Step 5. Determine the target point that the agents will move towards. The target point is calculated as $target_j = \sum_{i=1}^{N} x_{ij} * w_i$.

Step 6. Calculate the fitness of the target point, $target$.

Step 7. Determine the direction ($direct_i$) of agent i by using the position ($target$) and fitness value ($f(x_{target})$) of the target point.

Step 8. Update the position of each particle by $x_i = x_i + sl * direct_i * abs(x_i)$ where sl is an adaptive control parameter and abs is the absolute value function.

Step 9. Repeat the Steps 3-8 until a stopping condition is met.

3.5. Conceptual Comparison of EFO, EMA, GSA and WSA

EMA and EFO are meta-heuristic algorithms based on classical electromagnetic theory, whereas GSA is inspired from gravitational theory and WSA is based on superposition principle. In WSA, GSA and EMA a general resultant force is determined and each solution's position is updated according to the direction of the resultant force. On the other hand, EFO uses an intelligent EMP selection mechanism and new solutions are constructed using those selected solutions. All algorithms employ similar random components i.e. random step length or random mutation in order to enhance the exploration behaviour. Only EFO uses a greedy selection procedure in which candidate solutions that are worse than the worst solution in the current population are destroyed and not accepted. WSA, GSA and EMA exploit the information of the better solutions by giving more weight to those solutions. Instead, EFO employs direct information sharing by selection solution from positive field for each candidate generation.

4. COMPUTATIONAL RESULTS

4.1. Parameter Settings and Experimental Setup

Parameter settings of the algorithms may have a great influence on the performance. In general, the parameters of the algorithms are set to their original values given in their corresponding papers.

In order to compare algorithms objectively, population sizes are set to 50 for all algorithms and a 30 seconds CPU time is taken as the stopping criteria. For the EFO, the positive field is set to 10% of the total population, whereas negative field is taken as 45% $(p\_field = 0.10, \quad n\_field = 0.45)$. Further, ps_rate and r_rate are set as 0.2 and 0.3, respectively as given in [27]. For EMA, the only parameter is the population size and set to 50 as aforementioned. For GSA, G_0 and α are set to 100 and 20, respectively. K_0 is set to the population size [29]. For WSA, the control parameters τ, sl_0, φ, and λ are taken as 0.8, 0.035, 0.001 and 0.75, respectively [30].

All experiments were implemented in MATLAB 8.1 and executed on the same computer with Intel Xeon CPU (2.67 GHz) and 16 GB of memory. Further, all algorithms have been run 30 times with random seeds and the results are reported accordingly. It should be noted here that all calculations have been carried out on six decimal point base.

4.2. Benchmark problems

In order to investigate the effectiveness of the proposed algorithms, the standard $I-V$ dataset for a single diode cell from [3] is adopted. The dataset contains 26 samples and data has been achieved from the system under 1 sun $(1000\,W/m^2)$ at 33℃, where a commercial silicon solar cell (from the R.T.C. Company of France) with a diameter of 57 mm was used.

4.3. Computational Results

The best results out of 30 runs for each benchmark problem in terms of $f(V_t, I_t, \boldsymbol{X})$ $(I_t^M - I_t^C)$ are listed in Table 2. In Table 2, the best results for each data point are given in bold. From this table, it can be observed that EFO and EMA achieve similar results and outperform GSA and WSA. Both EFO and EMA obtain the best results on 13 cases out of 26.

In addition to the detailed results given in Table 2, the parameter values and RMSE for the best runs and the mean RMSE values over 30 runs are tabulated in Table 3. From Table 3, it can be seen that EFO is the best performing algorithm with a RMSE value of 9.860219E-04 and a mean RMSE of 9.86E-04. The mean RMSE values that are obtained by EMA, GSA and WSA are 1.04E-03, 3.63E-03 and 1.43E-01, respectively. On the other hand, the achieved parameter values are R_s*=0.036377*, $R_{sh} = 53.718646$, $I_{ph} = 0.760776$, $I_{sd} = 0.323022$ and $n = 1.481184$ for the best run with EFO.

Table 2. Terminal $(V_t - I_t)$ measurements and error values for the single diode model

			EFO		**EMA**		**GSA**		**WSA**	
Data	$V_t^M(V)$	$I_t^M(A)$	$I_t^C(A)$	$f(V_t, I_t, \boldsymbol{X})$	$I_t^C(A)$	$f(V_t, I_t, \boldsymbol{X})$	$I_t^C(A)$	$f(V_t, I_t, \boldsymbol{X})$	$I_t^C(A)$	$f(V_t, I_t, \boldsymbol{X})$
1	-0.2057	0.764000	0.764088	**-0.000088**	0.763710	0.000290	0.763163	0.000837	0.756338	0.007662
2	-0.1291	0.762000	0.762663	-0.000663	0.762368	**-0.000368**	0.762240	-0.000240	0.755556	0.006444
3	-0.0588	0.760500	0.761355	-0.000855	0.761136	**-0.000636**	0.761392	-0.000892	0.754838	0.005662
4	0.0057	0.760500	0.760154	**0.000346**	0.760004	0.000496	0.760612	-0.000112	0.754177	0.006323
5	0.0646	0.760000	0.759055	**0.000945**	0.758969	0.001031	0.759896	0.000104	0.753570	0.006430
6	0.1185	0.759000	0.758042	**0.000958**	0.758014	0.000986	0.759227	-0.000227	0.753000	0.006000
7	0.1678	0.757000	0.757092	**-0.000092**	0.757116	-0.000116	0.758575	-0.001575	0.752437	0.004563
8	0.2132	0.757000	0.756141	0.000859	0.756214	**0.000786**	0.757863	-0.000863	0.751808	0.005192
9	0.2545	0.755500	0.755087	0.000413	0.755201	**0.000299**	0.756948	-0.001448	0.750972	0.004528
10	0.2924	0.754000	0.753664	0.000336	0.753814	**0.000186**	0.755512	-0.001512	0.749625	0.004375
11	0.3269	0.750500	0.751391	**-0.000891**	0.751567	-0.001067	0.753000	-0.002500	0.747240	0.003260
12	0.3585	0.746500	0.747354	**-0.000854**	0.747545	-0.001045	0.748420	-0.001920	0.742886	0.003614
13	0.3873	0.738500	0.740117	**-0.001617**	0.740308	-0.001808	0.740302	-0.001802	0.735194	0.003306
14	0.4137	0.728000	0.727382	0.000618	0.727554	**0.000446**	0.726395	0.001605	0.722084	0.005916
15	0.4373	0.706500	0.706973	**-0.000473**	0.707109	-0.000609	0.704740	0.001760	0.701729	0.004771
16	0.4590	0.675500	0.675280	0.000220	0.675365	**0.000135**	0.672021	0.003479	0.671023	0.004477
17	0.4784	0.632000	0.630758	0.001242	0.630783	**0.001217**	0.627069	0.004931	0.628771	0.003229
18	0.4960	0.573000	0.571928	**0.001072**	0.571895	0.001105	0.568594	0.004406	0.573517	-0.000517
19	0.5119	0.499000	0.499607	-0.000607	0.499524	**-0.000524**	0.497362	0.001638	0.505601	-0.006601
20	0.5265	0.413000	0.413649	-0.000649	0.413532	**-0.000532**	0.413061	-0.000061	0.424327	-0.011327
21	0.5398	0.316500	0.317510	-0.001010	0.317378	**-0.000878**	0.318644	-0.002144	0.331990	-0.015490
22	0.5521	0.212000	0.212155	-0.000155	0.212028	**-0.000028**	0.214664	-0.002664	0.228690	-0.016690
23	0.5633	0.103500	0.102251	**0.001249**	0.102148	0.001352	0.105277	-0.001777	0.118134	-0.014634
24	0.5736	-0.010000	-0.008718	-0.001282	-0.008783	**-0.001217**	-0.006694	-0.003306	0.002643	-0.012643
25	0.5833	-0.123000	-0.125507	**0.002507**	-0.125520	0.002520	-0.125585	0.002585	-0.121856	-0.001144
26	0.5900	-0.210000	-0.208472	**-0.001528**	-0.208442	-0.001558	-0.211710	0.001710	-0.214073	0.004073

V_t^M: Measured terminal voltage I_t^M: Measured terminal current I_t^C: Calculated terminal current

Table 3. Comparison of the results for the single diode model

Item	**EFO**	**EMA**	**GSA**	**WSA**
$x_1: R_s(\Omega)$	0.036377	0.036365	0.032130	0.027957
$x_2: R_{sh}(\Omega)$	53.718646	57.025188	82.871489	97.854073
$x_3: I_{ph}(A)$	0.760776	0.760590	0.760977	0.754454
$x_4: I_{sd}(\mu A)$	0.323022	0.329155	0.847206	1.000000
$x_5: n$	1.481184	1.483019	1.585214	1.607072
$RMSE$	**9.860219E-04**	9.972880E-04	2.166195E-03	7.702232E-03
Mean	**9.86E-04**	1.04E-03	3.63E-03	1.43E-01

5. CONCLUSIONS

In this study, four different and well-known physics-inspired meta-heuristic algorithms are compared to identify their performances for the parameter estimation of photovoltaic cells. To compare the algorithms, a well-known benchmark data set introduced for a single diode model is used. In order to make a fair comparison, the initial forms of the algorithms are taken into account. Moreover, the stopping criterion of each algorithm is set to a maximum time limit of 30 seconds and the population sizes are taken as 50. As a result of the computations, the best found RMSE values obtained by the algorithms show that the EFO shows better performance and outperforms other three meta-heuristic algorithms by estimating the solar cell parameters with lower RMSE value.

Since this paper is the first study that considers the physics-inspired meta-heuristic algorithms for the SCPIP, this research can be extended with the following perspectives. In addition to the single diode model, the performance of the algorithms on the double diode model can be analysed. Another extension on the paper can be done by considering additional physics-inspired meta-heuristic algorithms. On the other hand, the performance of the meta-heuristic algorithms are sensitive to the control parameters of the algorithms. Therefore, parametric analysis for each algorithm may be carried out. Finally, the original versions of the algorithms can be improved to obtain more efficient and effective results for the SCPIP.

References

[1] Askarzadeh A., Rezazadeh A., Artificial bee swarm optimization algorithm for parameters identification of solar cell models, *Applied Energy*, Vol. 102, 2013, pp. 943-949.

[2] Yu K., Liang J.J., Qu B.Y., Chen X., Wang H., Parameters identification of photovoltaic models using an improved JAYA optimization algorithm, *Energy Conversion and Management*, Vol. 150, 2017, pp. 742-753.

[3] Easwarakhanthan T., Bottin J., Bouhouch I, Boutrit C., Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, *International Journal of Solar Energy*, Vol. 4, No. 1, 1986, pp. 1-12.

[4] Chan D.S.H., Phillips J.R., Phang J.C.H., A comparative study of extraction methods for solar cell model parameters, *Solid-State Electronics*, Vol. 29, No. 3, 1986, pp. 329-337.

[5] Ortiz-Conde A., Garcia sanchez F.J., Muci J., New method to extract the model parameters of solar cells from the explicit analytic solutions of their illuminated $I-V$ characteristics, *Solar Energy Materials and Solar Cells*, Vol. 90, No. 3, 2006, pp. 352-361.

[6] Saleem H., Karmalkar S., An analytical method to extract the physical parameters of a solar cell from four points on the illuminated $J-V$ curve, *IEEE Electron Device Letters*, Vol. 30, No. 4, 2009, pp. 349- 352.

[7] Askarzadeh A., Rezazadeh A., Parameter identification for solar cell models using harmony search-based algorithms, *Solar Energy*, Vol. 86, No. 11, 2012, pp. 3241-3249.

[8] El-Naggar K.M., AlRashidi M.R., AlHajiri M.F., Al-Othman A.K., Simulated annealing algorithm for photovoltaic parameters identification, Solar Energy, Vol. 86, No. 1, 2012, pp. 266-274.

[9] Jervase J.A., Bourdoucen H., Al-Lawati A., Solar cell parameter extraction using genetic algorithms, *Measurement Science and Technology*, Vol. 12, No. 11, 2001, pp. 1922-1925.

[10] Zagrouba M., Sellami A., Bouaicha M., Ksouri M., Identification of PV solar cells and modules parameters using the genetic algorithms: Application to maximum power extraction, *Solar Energy*, Vol. 84, No. 5, 2010, pp. 860-866.

[11] Ye M., Wang X., Xu Y., Parameter extraction of solar cells using particle swarm optimization, *Journal of Applied Physics*, Vol. 105, 2009, pp. 1-9.

[12] Ishaque K., Salam Z., An improved modelling method to determine the model parameters of photovoltaic (PV) modules using differential evolution (DE), *Solar Energy*, Vol. 85, No. 9, 2011, pp. 2349-2359.

[13] Jiang L.L., Maskell D.L., Patra J.C., Parameter estimation of solar cells and modules using an improved adaptive differential evolution algorithm, *Applied Energy*, Vol. 112, 2013, pp. 185-193.

[14] Gong W., Cai Z., Parameter extraction of solar cell models using repaired adaptive differential evolution, *Solar Energy*, Vol. 94, 2013, pp. 209-220.

[15] Oliva D., Cuevas E., Pajares G., Parameter identification of solar cells using artificial bee colony optimization, *Energy*, Vol. 72, 2014, pp. 93-102.

[16] Rajasekar N., Kumar N.K., Venugopalan R., Bacterial foraging algorithm based solar PV parameter estimation, *Solar Energy*, Vol. 97, 2013, pp. 255-265.

[17] Awadallah M.A., Variations of the bacterial foraging algorithm for the

extraction of PV module parameters from nameplate data, *Energy Conversion and Management*, Vol. 113, 2016, pp. 312-320.

[18] Niu Q., Zhang H., Li, K., An improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models, *International Journal of Hydrogen Energy*, Vol. 39, 2014, pp. 2837-3854.

[19] Patel S.J., Panchal A.K., Kheraj V., Extraction of solar cell parameters from a single current-voltage characteristic using teaching learning based optimization algorithm, *Applied Energy*, Vol. 119, 2014, pp. 384-393.

[20] Chen X., Yu K., Du W., Zhao W., Liu G., Parameters identification of solar cell models using generalized oppositional teaching learning based optimization, *Energy*, Vol. 99, 2016, pp. 170-180.

[21] Niu Q., Zhang H., Li, K., A biogeography-based optimization algorithm with mutation strategies for model parameter estimation of solar and fuel cells, *Energy Conversion and Management*, Vol. 86, 2014, pp. 1173-1185.

[22] Askarzadeh A., Coelho L.dos S., Determination of photovoltaic modules parameters at different operating conditions using a novel bird mating optimizer approach, *Energy Conversion and Management*, Vol. 89, 2015, pp. 608-614.

[23] Jacob B., Balasubramanian K., Babu S., Azharuddin S.M., Rajasekar N., Solar PV modelling and parameter extraction using artificial immune system, *Energy Procedia*, Vol. 75, 2015, pp. 331-336.

[24] Guo L., Meng Z., Sun Y., Wang L., Parameter identification and sensitivity analysis of solar cell models with cat swarm optimization algorithm, Energy Conversion and Management, Vol. 108, 2016, pp. 520-528.

[25] Allam D., Yousri D.A., Eteiba M.B., Parameters extraction of the three diode model for the multi-crystalline solar cell/module using Moth-Flame optimization algorithm, *Energy Conversion and Management*, Vol. 123, 2016, pp. 535-548.

[26] Oliva D., El Aziz M.A., Hassanien A.E., Parameter estimation of photovoltaic cells using an improved chaotic whale optimization algorithm, *Applied Energy*, Vol. 200, 2017, pp. 141-154.

[27] Abedinpourshotorban H., Shamsuddin S. M., Beheshti Z., Jawawi D.N., Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm, *Swarm and Evolutionary Computation*, 26, 2016, pp. 8-22.

[28] Birbil I, Fang S.-C., An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, Vol. 25, No. 3, 2003, pp. 263-282.

[29] Rashedi E., Nezamabadi-Pour H., Saryazdi S., GSA: A gravitational search algorithm, *Information Sciences*, Vol. 179, No. 13, 2009, pp. 2232-2248.

[30] Baykasoğlu A., Akpinar Ş., Weighted Superposition Attraction (WSA): A swarm intelligence algorithm for optimization problems–Part 1: Unconstrained optimization, *Applied Soft Computing*, Vol. 56, 2017, pp. 520-540.

[31] Özbakır L., Turna F., Clustering performance comparison of new generation meta-heuristic algorithms, *Knowledge-Based Systems*, Vol. 130, 2017, pp. 1-16.