Neurocomputing ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

freepaper.me FREE paper

A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm

Hussain Shareef [a], Ammar Hussein Mutlag [b,c,*], Azah Mohamed [b]

[a] TNB Research Sdn. Bhd., No. 1, Lorong Ayer Itam Kawasan Institusi Penyelidikan, Kajang 43000, Selangor, Malaysia
[b] Department of Electrical, Electronic and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600 Bnagi, Selangor, Malaysia
[c] College of Electrical and Electronic Techniques, Foundation of Technical Education, Baghdad, Iraq

ARTICLE INFO

Article history:
Received 24 November 2014
Received in revised form
17 April 2015
Accepted 23 May 2015
Communicated by Cheng-Wu Chen

Keywords:
Lightning search algorithm (LSA)
Nature-inspired algorithms
Fuzzy logic controller (FLC)
Inverter
Photovoltaic (PV)

ABSTRACT

Photovoltaic (PV) inverters convert DC voltage and current to AC quantities whose magnitude and frequency are controlled to obtain the desired output. While there are plenty of controllers, it is the fuzzy logic controller (FLC) that receives increasing attention. In this study, a novel metaheuristic optimization algorithm known as lightning search algorithm (LSA) is presented for solving the problem of trial and error procedure in obtaining membership functions (MFs) used in the conventional FLCs. The LSA mimics the natural phenomenon of lightning. It is generalized from the mechanism of step leader propagation. The proposed optimization algorithm considers the concept of fast particles known as projectiles. The probabilistic nature and tortuous characteristics of lightning discharges, which depend on the type of projectile, are modeled using various random distribution functions. To evaluate the reliability and efficiency of the proposed algorithm, the LSA is first tested using 10 benchmark functions with various characteristics necessary to evaluate a new algorithm. Then it is used in designing optimum FLC for standalone PV inverter. The result demonstrates that the LSA generally provides better outcome compared with the other tested methods with a high convergence rate.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Photovoltaic (PV) system has played very important role in the renewable energy (RE) technologies because PV systems are environment friendly, clean, and secure energy sources [1]. The PV based RE technologies has received a lot of attentions for stand-alone (SA) and grid-connected (GC) systems [2]. In the SA mode the inverter should be able to generate high quality power to the loads. An inverter is required to connect the load to the PV where the latter generates DC power [3]. The output waveforms, voltage and current, under the SA mode of operation of the inverter should be controlled based on the reference values. Therefore, a voltage source inverter (VSI) and a suitable voltage control approach are required [4]. The main feature of a good power inverter is its capability to provide constant amplitude sinusoidal voltage and frequency regardless the typing of the load it is connected to. The power inverter must also have the capability to quickly recover from transients affected by the disturbances without causing power quality problems. However, the large-scale use of PV generators raises many challenges, such as harmonic pollutions, low efficiency of energy conversion, fluctuation of output power, and reliability of power electronic converters [5].

Various inverter control techniques have been suggested by many researchers and discussed in [6]. FLCs have become increasingly popular in designing inverter controls because it include an advantage over the traditional controller by reducing the dependence on the mathematical model [7,8]. Nonetheless, the performance of FLCs depends on the rule basis, number of rules, and MFs. These variables are determined by a trial and error procedure, which is time consuming [9,10]. Thus, to overcome these limitations in FLC design, an optimization method was suggested in [11]. The method uses particle swarm optimization (PSO) algorithm and FLC for maximum power point tracking. However, PSO is prone to premature convergence and hence there is a need to find a better way to optimize the FLC. In a related work [12–14], the problem of adaptive fuzzy tracking control for a class of uncertain multiple-input–multiple-output pure-feedback nonlinear systems with immeasurable states was considered. In this work, fuzzy logic systems were first used to approximate the unknown nonlinear functions, and then a fuzzy state observer was designed to estimate the unmeasured states.

Many techniques have been suggested to deal with optimization problems. The optimum solution in most classical point-by-point methods is obtained by deterministic procedure [15]. As the

* Corresponding author.
E-mail addresses: hussain.shareef@tnbr.com.my (H. Shareef), ammar_alqiesy@yahoo.com (A.H. Mutlag), azah@eng.ukm.my (A. Mohamed).

http://dx.doi.org/10.1016/j.neucom.2015.05.083
0925-2312/© 2015 Elsevier B.V. All rights reserved.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

size of the search space increase with the dimension of the optimization problem; the finding of the optimum solution for such problems using classical techniques becomes complicated [16]. Recently, computational intelligence optimization algorithms have been extensively used to solve complex optimization problems in various domains, including science, commerce, and engineering, because of their ease of use, broad applicability, and global perspective. computational-intelligence means search of solution for a problem by natural or artificial agents that address complex real-world problems [17]. These algorithms can be further divided into swarm intelligence methods and evolutionary algorithms (EAs). Swarm intelligence optimization algorithms generally use reduced mathematical models of the complex social behavior of insect or animal groups.

The most popular swarm intelligence methods are particle swarm optimization (PSO) [18], artificial bee colony (ABC) [19], and ant colony optimization (ACO) [20]. The PSO mimics the movements of bird flocking or fish schooling [21]. Inspired by the food-searching mechanism of honey bees, the ABC method uses the foraging behavior of these insects [22]. Meanwhile, ACO was developed based on the behavior of ants when seeking the optimal path between their colony and food source [20]. However, these swarm intelligence methods are limited by factors such as trapping in local minima and premature convergence [21–23]. To overcome these problems, variants of these algorithms have been developed with superior performance [21,23–25]. Other swarm intelligence methods, such as the gravitational search algorithm (GSA) [26], the harmony search algorithm [27], biogeography-based optimization [28], and the grenade explosion method [29], have also been developed.

EAs derive their working principles from natural genetic evolution. At each generation, the best individuals of the current population survive and produce offspring resembling them; hence, the population gradually comprises enhanced individuals. Operations such as recombination, crossover, mutation, selection, and adaptation are involved in this process [30]. The renowned paradigms of EAs are the genetic algorithm (GA) [30], evolutionary programming [31], differential evolution [32], evolutionary strategy [28], and genetic programming [33]. These algorithms are based on the principles of Darwinian theory and evolution theory of living beings. However, each algorithm follows specialized recombination, crossover, mutation, selection, and adaptation strategies. Similar to other metaheuristic algorithms, the aforementioned methods also have some drawbacks, such as slow convergence rate, difficulty in solving multimodal functions, and stagnation in local minima [34–36]. Advanced versions of EAs have been developed in recent years to improve the efficiency and performance of the aforementioned EAs; these advanced algorithms include stud genetic algorithm [36], fast evolutionary programming [35], adaptive differential evolution algorithm [37], and covariance matrix adaptation evolution strategy [38]. Not all algorithms and their variants provide superior solutions to some specific problems. Therefore, new heuristic optimization algorithms must be continuously searched to advance the field of computational intelligence optimization.

This study aims to introduce a novel metaheuristic optimization method called the lightning search algorithm (LSA) to solve the problem of trial and error procedure in obtaining membership functions (MFs) used in conventional FLCs. The LSA is based on the natural phenomenon of lightning. The proposed optimization algorithm is generalized from the mechanism of step leader propagation. It considers the involvement of fast particles known as projectiles in the formation of the binary tree structure of a step leader. Three projectile types are developed to represent the transition projectiles that create the first step leader population *N*, the space projectiles that attempt to become the leader, and the lead projectile that represent the best positioned projectile originated among *N* number of step leaders. The probabilistic nature and tortuous characteristics of lightning discharges, which depend on the type of projectile, are modeled using various random distribution functions. The utilization of the LSA is expected to improve the performance of FLCs for PV inverters. For this purpose, a unique objective function which considers the mean square error (MSE) of the three-phase PV inverter output voltage was formulated. The LSA aims to minimize MSE by adaptively tuning the MFs of FLC during the controller design phase.

2. Mechanism of lightning

Lightning is a fascinating and impressive natural phenomenon. The probabilistic nature and tortuous characteristics of lightning discharges originate from a thunderstorm. Among the common displays of lightning, downward negative cloud-to-ground lightning flashes (Fig. 1) are the most studied event in lightning research [39].

During a thunderstorm, charge separation occurs within the cloud, usually with a negative charge below and a positive charge above. This process creates a strong electric field. Free electrons generated by cosmic radiation or natural radioactivity are generally attached to oxygen molecules to form negative ions [40]. However, under high electric fields, a part of the electron velocity distribution possesses enough energy to ionize air, thereby generating additional electrons together with the original ones.

When one of these free electrons is accelerated by the electric field in the region where ionization probability is higher than attachment probability, an electron avalanche occurs. This phenomenon eventually causes a negative corona streamer. Nonetheless, the streamers are weak and become electrically isolated from their point of origin when the ionization probability is lower than the attachment probability.

The so-called streamer-to-leader transition happens at high electron densities and temperatures when narrow channels are thermalized. The negative or the step leader's movement from cloud to ground is not continuous but progresses through regular and discrete steps. For the leader to progress, a leader section in the vicinity in front of the old leader channel, called a space leader, develops from the preceding corona streamers. The space leader propagates backward until it connects to the old leader channel forming a long channel with a tip having the same electric potential [41]. A current wave is generated during this process.

Once this current wave reaches the tip of the new leader, a burst of corona streamers again propagates out in front of the new tip. Afterward, a new space leader originates in front of the current leader tip. The procedure is then repeated. The development is a

Fig. 1. Step leaders descending from a storm cloud.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

stochastic growth process in which the probability of a step to grow in a certain direction is determined by the state of the medium and the projection of the local electric field in this direction [39,42].

As the step leaders approach the earth, high-density negative charges at the leader tip stimulate a concentrated positive charge on earth. When the electric field intensity resulting from the positive charges near the ground becomes large enough, upward positive streamers from the ground are initiated, and the step leader bridges the gap as they move their way down. This phenomenon determines the lightning strike point and the main lightning current path between the cloud and the ground. The step leader is not the lightning strike; it only maps the optimal course that the strike will follow.

3. Lightning search algorithm (LSA)

The proposed optimization algorithm is generalized from the hypothesis presented in [43] as a mechanism of step leader propagation. It considers the involvement of fast particles known as projectiles in the formation of the binary tree structure of the step leader and in the concurrent formation of two leader tips at fork points instead of a conventional step leader mechanism that uses the concept of streamers described above.

3.1. Projectile and step leader propagation

Hydrogen, nitrogen, and oxygen atoms can be found near the region of thunderclouds. During the intensive freezing of water molecules within a thundercloud, a fraction of the water molecules incapable to fit in the ice structure are squeezed out of the forming ice at intense speeds. The oxygen and hydrogen atoms are separated during the process and ejected in a random direction as projectiles. It travels through the atmosphere after being ejected from the thunder cell, and provides an initial ionization path or channel through collision and transition to the step leader. In the proposed algorithm, each projectile from the thunder cell is assumed to create a step leader and a channel. In other words, the projectile represents the initial population size. The concept of projectile is highly similar to the term “particle” or “agent” used in PSO and the GSA, respectively. The projectiles suggest random solutions for corresponding problems to be solved by the LSA. In the present study, the solution refers to the tip of the current step leader's energy *Ec*.

3.2. Properties of the projectile

A projectile that moves in the atmosphere under normal conditions loses its kinetic energy during elastic collisions with molecules and atoms in the air. The velocity for the projectile is given as

$$v_p = \left[1 - \left(1/\sqrt{1-(v_0/c)^2} - sF_i/mc^2\right)^{-2}\right]^{-1/2} \quad (1)$$

where v_p and v_0 are the current velocity and initial velocity, respectively, of the projectile; c is the light speed; F_i is the constant ionization rate; m is the mass of the projectile; and s is the length of the path traveled.

Eq. (1) clearly shows that velocity is a function of leader tip position and projectile mass. When the mass is small or when the path traveled is long, the projectile has little potential to ionize or explore a large space. It can only ionize or exploit the nearby space. Therefore, the exploration and exploitation abilities of the algorithm can be controlled by using the relative energies of the step leaders.

Another important property of a stepped leader is forking, in which two simultaneous and symmetrical branches emerge. This phenomenon rarely occurs because of nuclei collision. Any additional channel created during forking increases the number of projectiles by one and hence the population size. In the proposed algorithm, forking is realized in two ways. First, symmetrical channels are created because the nuclei collision of the projectile is realized by using the opposite number as in Eq. (2) [44].

$$\overline{p}_i = a + b - p_i \quad (2)$$

where $\overline{p}_i$ and p_i are the opposite and original projectiles, respectively, in a one-dimensional system; a and b are the boundary limits. This adaptation may improve some of the bad solutions in the population. If forking does not improve channel propagation in the LSA, one of the channels at the forking point is illuminated to maintain the population size.

In the second type of forking, a channel is assumed to appear at a successful step leader tip because of the energy redistribution of the most unsuccessful leader after several propagation trials. The unsuccessful leader can be redistributed by defining the maximum allowable number of trials as channel time. In this case, the population size of step leaders does not increase.

3.3. Projectile modeling and step leader movement

Three projectile types are developed to represent the transition projectiles that create the first-step leader population *N*, the space projectiles that attempt to reach the best leader position, and the lead projectiles that represent the best position among *N* numbers of step leaders. In this case, a one-dimensional projectile type is illustrated for clarity.

3.3.1. Transition projectile

As mentioned previously, a leader tip is formed at an early stage because the transition forms an ejected projectile from the thunder cell in a random direction. Therefore, it can be modeled as a random number generated from the standard uniform probability distribution on the open interval representing the solution space. The probability density function $f(x^T)$ of the standard uniform distribution can be represented as

$$f(x^T) = \begin{cases} 1/b-a & \text{for} \quad a \le x^T \le b \\ 0 & \text{for} \quad x < a \ \text{ or } \ x^T > b \end{cases} \quad (3)$$

where x^T is a random number that may provide a solution or the initial tip energy E_{sl_i} of the step leader sl_i; a and b are the lower and upper bounds, respectively, of the solution space. For a population of N step leaders $SL = [sl_1, sl_2, sl_3, ..., sl_N]$, N random projectiles $P^T = [p_1^T, p_2^T, p_3^T, ..., p_N^T]$ that satisfy the solution dimension are required.

3.3.2. Space projectile

Once the N step leader tips are evolved, the leaders need to move using energetic projectiles by ionizing the section in the vicinity of the old leader tip in the next step *step+1*. The position of the space projectile $P^S = [p_1^S, p_2^S, p_3^S, ..., p_N^S]$ at *step+1* can be partially modeled as a random number drawn from the exponential distribution with shaping parameter μ. The probability density function $f(x^S)$ of an exponential distribution is given by

$$f(x^S) = \begin{cases} \frac{1}{\mu} e^{\frac{-x^S}{\mu}} & \text{for} \quad x^S \ge 0 \\ 0 & \text{for} \quad x^S \le 0 \end{cases} \quad (4)$$

Eq. (4) shows that the space projectile position or the direction in the next step can be controlled by shaping parameter μ. In the LSA,

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

Algorithm: Lightning Search Algorithm

N="The size of the population", **D**= dim "Number of dimension", **T**= "Maximum number of iteration"

Dpoint = initialize( )
channel = 0;
channel_max = 10;
fit_old = 10^{10} . [ones(1,N)]
direct = sign [unifrnd (-1,1,1,dim)]
for t = 1:T **do**
  fit = evaluateF (Dpoint , F_index)
  E = fit
  channel = channel +1
  if channel >= channel_max **then**
    [Ms ds]=sort (E , 'ascend')
    $Dpoint_{ds(N)} := Dpoint_{ds(1)}$:
    E [ds(N)] = E [ds(1)]
    channel = 0
  endif
  [Ms ds]=sort (E,'ascend')
  best = $E_{ds(1)}$
  worst = $E_{ds(N)}$
  Energy = $f(_T)$
  for d = 1:D **do**
    $Dpoint\_test = Dpoint_{ds(1)}$:
    $Dpoint\_test_d = Dpoint\_test_d + direct_d . 0.005 . [UB_d - LB_d]$
    fv_test = evaluateF(Dpoint_test,F_index)
    $direct_d = \begin{cases} direct_d & if\ fv\_test < best \\ -\ direct_d & else \end{cases}$
  endfor
  for i = 1:N **do**
    dist = $Dpoint_i$: - $Dpoint_{ds(1)}$:
    for d = 1:D **do**
      if $Dpoint_i$: == $Dpoint_{ds(1)}$: **then**
        $Dpoint\_temp_d = Dpoint_{i,d} + direct_d . abs [normrnd(0,Energy)]$
      else
        $Dpoint\_temp_d = \begin{cases} Dpoint_{i,d} + exprnd [abs(dist_d)] & if\ dist_d < 0 \\ Dpoint_{i,d} - exprnd (dist_d) & else \end{cases}$
      endif
      if $[Dpoint\_temp_d > UB_d]$ or $[Dpoint\_temp_d < LB_d]$ **then**
        $Dpoint\_temp_d = rand(1) . [UB_d - LB_d] + LB_d$
      endif
    endfor
    fv = evaluateF (Dpoint_temp , F_index)
    if $fv < E_i$ **then**
      $Dpoint_i := Dpoint\_temp$
      $E_i = fv$
      if rand < 0.01 **then**
        for d = 1:D **do**
          $Dpoint\_fock_d = UB_d + LB_d - Dpoint\_temp_d$
        endfor
        fock_fit = evaluateF (Dpoint_fock , F_index)
        if fock_fit < E_i **then**
          $Dpoint_i := Dpoint\_fock$
          E_i = fock_fit
        endif
      endif
    endif
  endfor
  if best == worst **then**
    break
  endif
  $fitness_t = min(fit)$
endfor
optimal = min(fitness)

Fig. 2. Pseudo-code of the proposed LSA algorithm.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

μ_i for a specific space projectile p_i^S is taken as the distance between the lead projectile p^L and the space projectile p_i^S under consideration. With this definition, the position of p_i^S at $step+1$ can be written as

$$p_{i\_new}^S = p_i^S \pm \text{exprand}(\mu_i) \tag{5}$$

where exprand is an exponential random number. If p_i^S is negative, then the generated random number should be subtracted because Eq. (4) only provides positive values. However, the new position $p_{i\_new}^S$ does not guarantee stepped leader propagation or channel formation unless the projectile energy $E_{p\_i}^S$ is greater than the step leader E_{sl_i} to extend the channel or until a good solution is found. If $p_{i\_new}^S$ provides a good solution at $step+1$, then the corresponding stepped leader sl_i is extended to a new position sl_{i_new}, and p_i^S is updated to $p_{i\_new}^S$. Otherwise, they remain unchanged until the next step. If $p_{i\_new}^S$ extends sl_{i_new} beyond the recent, most extended leader during this process, then it becomes the lead projectile.

3.3.3. Lead projectile

Presumably, the step leader that has traveled nearest to the ground and the projectile associated with it do not have enough potential to ionize large sections in front of the leader tip. Therefore, the lead projectile can be modeled as a random number generated by the standard normal distribution with the shape parameter μ and the scale parameter σ. The normal probability density function $f(x^L)$ is expressed as

$$f(x^L) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x^L-\mu)^2}{2\sigma^2}} \tag{6}$$

Eq. (6) shows that the randomly generated lead projectile can search in all directions from the current position defined by the shape parameter. This projectile also has an exploitation ability defined by the scale parameter. In the LSA, μ_L for the lead projectile p^L is taken as p^L, and the scale parameter σ_L exponentially decreases as it progresses toward the earth or as it finds the best solution. With this definition, the position of p^L at $step+1$ can be written as

$$p_{new}^L = \text{normrand}(\mu_L, \sigma_L) \tag{7}$$

where normrand is a random number generated by the normal distribution function. Similarly, the new lead projectile position p_{new}^L does not guarantee step leader propagation unless the projectile lead projectile energy $E_{p\_i}^L$ is greater than the step leader E_{sl_i} to extend to a satisfactory solution. If p_{new}^L provides a good solution at $step+1$, then the corresponding step leader sl_i is extended to a new position sl_{L_new}, and p^L is updated to p_{new}^L. Otherwise, they remain unchanged until the next step, as in the case of the space projectile. The whole procedure of the LSA is summarized in pseudo-code as shown in Fig. 2.

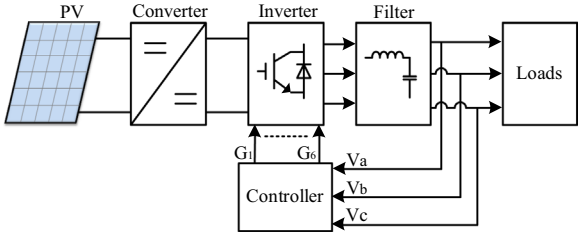

Fig. 3. Structure circuit of the three-phase inverter system.

4. Inverter control methodology

Inverter control is aimed at regulating the AC output voltage at a desired magnitude and frequency with low harmonic distortion. This regulation is carried out by the controller by implementing a proper control methodology to keep the voltage at a set reference. The configuration for the standalone PV inverter utilized in this study is presented in Fig. 3.

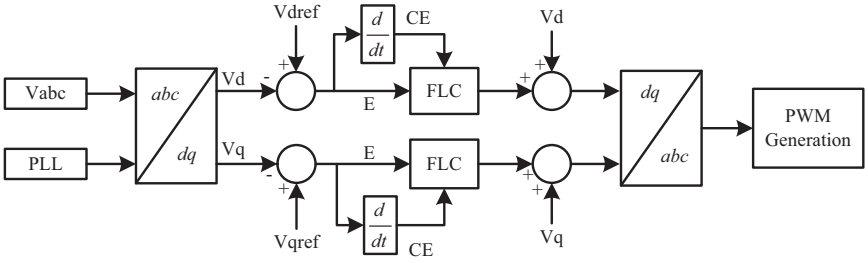

Fig. 4. Architecture of the voltage control strategy.

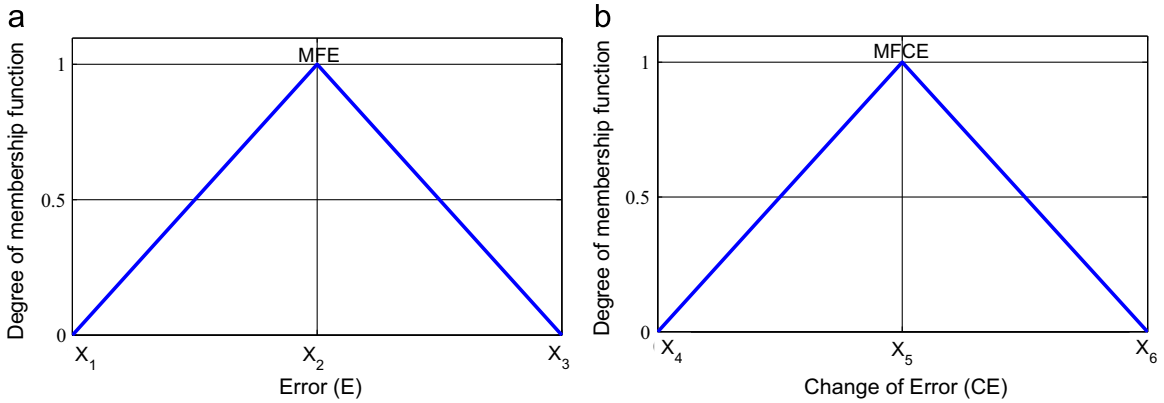

Fig. 5. Encoding of a MF.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

4.1. Control design procedure

The concept of the inverter control shown in Fig. 3 is explained in Fig. 4. To apply the strategy of the control in the inverter system, three-phase output voltages (V_a, V_b, and V_c) in the synchronous reference frame must be sensed at the load terminals. These voltages (V_a, V_b, and V_c) are then scaled and transformed into a d–q reference frame (V_d and V_q) to simplify the calculations for controlling the three-phase inverter [45,46].

Considering the non-linearity of the power conversion process of PV inverters, fuzzy logic is a convenient method to adopt in a PV inverter control system. The FLC represents the human expert decision in the problem solving mechanism. As shown in Fig. 4, the FLC have two inputs (E and CE). The first input (E) represents the error between the measured voltages V_d and V_q and reference voltages V_{dref} and V_{qref} per unit. The second input (CE) represents the change in error CE which is determined by taking the derivative of E. These signals (i.e., E and CE) are then sent to the controller at each sampling time T_s to compute the missing components in V_d and V_q and to generate the new V_d and V_q signals. The FLC design must pass through the following four steps [47,48].

4.1.1. Definition of the module characteristics

This is an important step to locate where the fuzzy logic controller can be located in the system which it help the designer to define the range for the inputs and output. In this work where FLC serves as a PV inverter controller, E and CE are used as inputs, whereas the missing component of V_d (or V_q) defined as O is used as the output of the FLC. For example, the two inputs for the FLC depicted in Fig. 4, namely, E and CE, at the tth sampling step corresponding to V_d can be represented as

$$E(t) = V_{dref} - V_d(t) \tag{8}$$

$$CE(t) = E(t) - E(t-1) \tag{9}$$

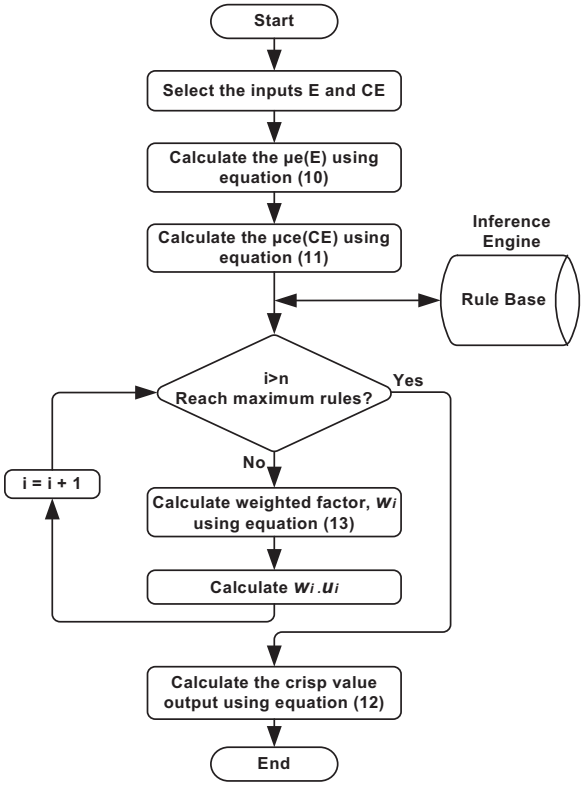

Fig. 6. Encoding of the FLC.

The output O for this case can be obtained at the last stage of the FLC design, which is clarified in the next section. After the inputs and output are defined, the next stage involves the fuzzification of inputs.

4.1.2. Fuzzifier design

This step performs the fuzzification interface. It translates the crisp values of "E" and "CE" as the fuzzy set "e" and "ce", respectively, through the MF degrees $\mu_e(E)$ and $\mu_{ce}(CE)$, which range from 0 to 1, as shown in Fig. 5 for triangular MFs. In Fig. 5(a), the membership function of error (MFE) is defined by three elements namely, X_1, X_2, and X_3 whereas the membership function of change of error (MFCE) in Fig. 5(b) is defined by another three elements represented as X_4, X_5, and X_6.

After defining the MFs, $\mu_e(E)$ and $\mu_{ce}(CE)$ can be expressed respectively as follows:

$$\mu_e(E) = \begin{cases} \frac{E - X_1}{X_2 - X_1} & X_1 \le E < X_2 \\ 1 + \frac{X_2 - E}{X_3 - X_2} & X_2 \le E < X_3 \end{cases} \tag{10}$$

$$\mu_{ce}(CE) = \begin{cases} \frac{CE - X_4}{X_5 - X_4} & X_4 \le CE < X_5 \\ 1 + \frac{X_5 - CE}{X_6 - X_5} & X_5 \le CE < X_6 \end{cases} \tag{11}$$

In a standard FLC design, the selection of the number of MFs and boundary values of each MF must be adjusted by the designer using the trial and error method until the FLC provides a satisfactory result. However, this process is time consuming and

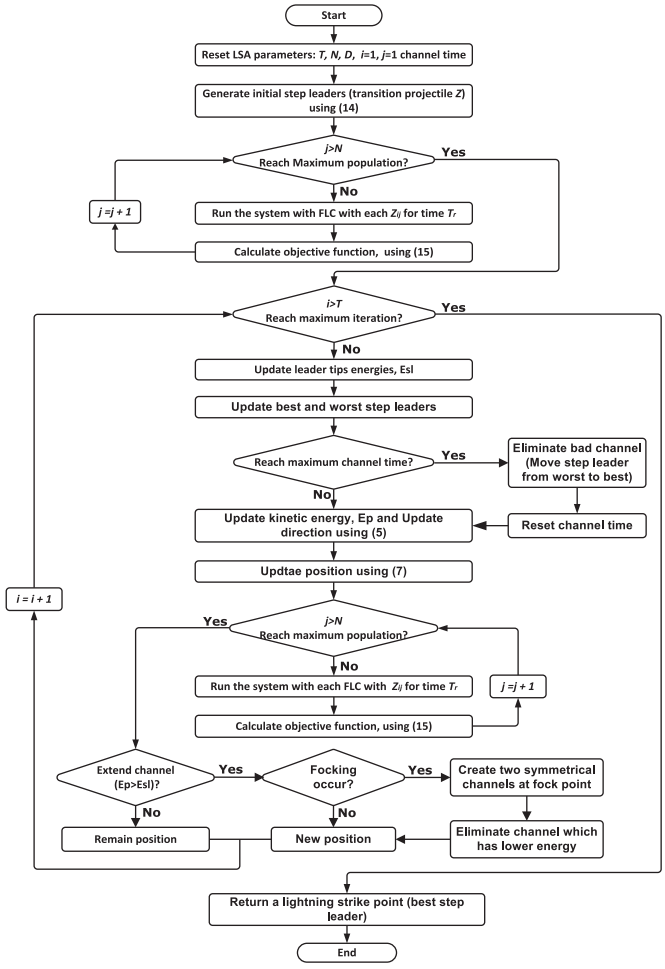

Fig. 7. Proposed LSA based optimum FLC design procedure.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

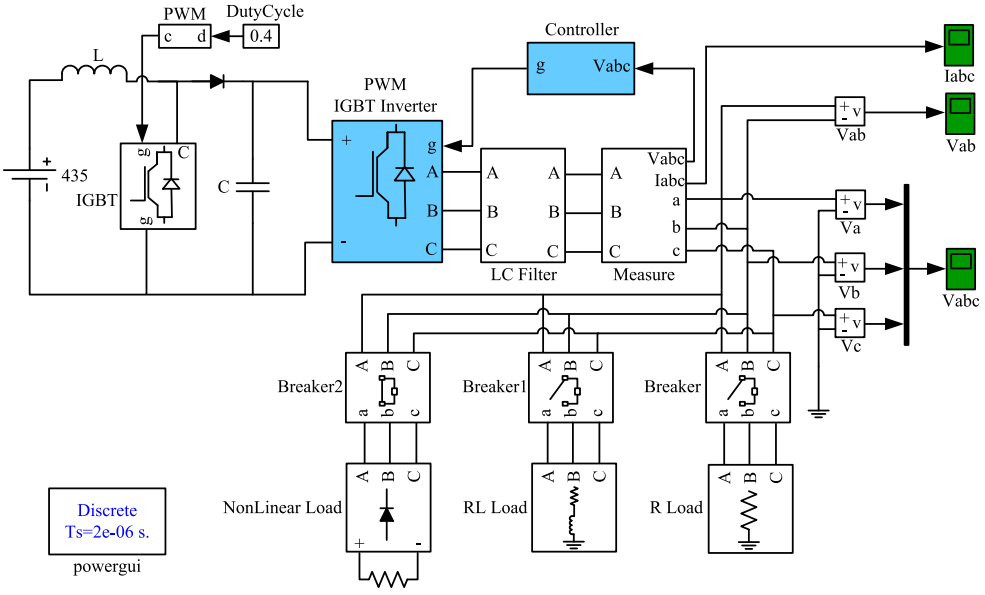

Fig. 8. Simulation model of the three phase inverter.

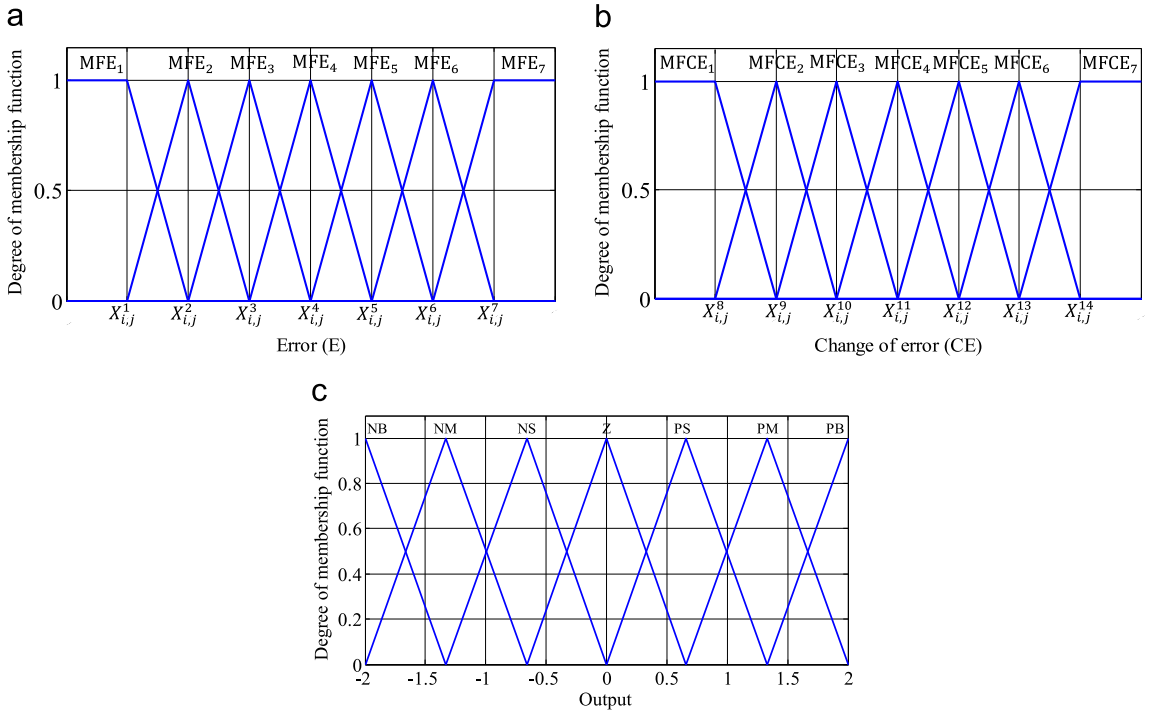

Fig. 9. FLC with 7 MFs for (a) *E* (b) *CE*, and output.

laborious. After the inputs are fuzzified, the fuzzy inputs are subjected to an inference engine for generating the fuzzy output.

4.1.3. Inference engine design

This step includes generating the rules in the form of IF (condition) THEN (action). The rules with two inputs for the Mamdani type are written as

R: IF *E* is "*label*" AND *CE* is "*label*", THEN *u* is "*label*".

The quantity of rules relates to the number of the inputs and MFs utilized in the FLC [47,48]. An FLC with large rule base demands a greater computational effort in terms of memory and computation time.

Table 1
Fuzzy control rules based on seven MF.

Error (*E*)	Change of error (*CE*)						
	$MFCE_1$	$MFCE_2$	$MFCE_3$	$MFCE_4$	$MFCE_5$	$MFCE_6$	$MFCE_7$
MFE_1	NB	NB	NB	NB	NM	NS	ZO
MFE_2	NB	NB	NB	NM	NS	ZO	PS
MFE_3	NB	NB	NM	NS	ZO	PS	PM
MFE_4	NB	NM	NS	ZO	PS	PM	PB
MFE_5	NM	NS	ZO	PS	PM	PB	PB
MFE_6	NS	ZO	PS	PM	PB	PB	PB
MFE_7	ZO	PS	PM	PB	PB	PB	PB

NB=negative big, NM=negative medium, NS=negative small, ZO=zero, PS=positive small, PM=positive medium, and PB=positive big.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

4.1.4. Defuzzifier design

The last step of the FLC is defuzzification. There are many approaches to perform the defuzzification. In this study, the center of area (COA) method given in (12) is used to generate the crisp value because it is more accurate and recommended [47].

$$O = \frac{\sum_{i=1}^{n} w_i.u_i}{\sum_{i=1}^{n} w_i} \tag{12}$$

where n is the number of rules, and w_i is the weighted factor that can be calculated using the Mamdani-*MIN* between $\mu_e(E)$ and $\mu_{ce}(CE)$ as expressed in (13).

$$w_i = MIN[\mu_e(E), \mu_{ce}(CE)] \tag{13}$$

The steps to implement the standard FLC are illustrated in Fig. 6.

Table 2
Parameter settings used in LSA, DSA, and PSO.

Parameter	LSA	DSA	PSO
Population Size	50	50	50
Max. Iteration	100, 200, 500	100, 200, 500	100, 200, 500
c1 and c2	–	–	2
p1 and p2	–	0.3.rand	–
Channel time	10	–	–

Table 3
Unimodal and separable test functions.

Function ID	Name	Expression	n	Search space	Function minimum
F1	Sphere	$f_1(\mathbf{x}) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0
F2	Quartic	$f_2(\mathbf{x}) = \sum_{i=1}^{n} ix_i^4 + \text{rand}(0,1)$	30	$[-1.28,1.28]^n$	0

Table 4
Test 1 global optimization results for benchmark functions in Table 3.

Function ID	Statistics	LSA	DSA	PSO
F1	Best	**1.06220E−19**	0.939999414	1.76307E−06
	Worst	**2.40506E−06**	37.18006813	0.000281909
	Average	**4.81067E−08**	11.58475565	2.76248E−05
	Median	**1.86102E−15**	9.501717692	1.67426E−05
	Standard deviation	**3.40126E−07**	6.938443045	4.32130E−05
F2	Best	**0.016268885**	0.044973120	95.86581258
	Worst	**0.040825354**	0.359368482	186.9633517
	Average	**0.024079674**	0.123075150	138.8343114
	Median	**0.022692902**	0.104513507	135.5998742
	Standard deviation	**0.005726198**	0.065346271	22.07744869

4.2. Proposed optimum FLC design procedure

As noted in the FLC design procedure, the main drawback of FLC design is the time-consuming trial and error process used to adjust the boundary values of MFs in the fuzzification process. An improper selection of MF boundaries may lead to the poor performance of the overall system. Therefore, this paper presents a methodology to optimize MFs using a new heuristic optimization algorithm. The heuristic optimization algorithms are used for solving complex and intricate problems which are otherwise difficult to solve by classical methods. They are population based methods designed to solve a problem more quickly or to find an approximate solution when classic methods fail to find the solution especially with multimodal optimization problems. The LSA is used as a heuristic optimization tool for adjusting the boundary values of MFs adaptively of the FLC design for PV inverter control. The four basic components that are vital to any optimization method are input vectors, objective function formulation, optimization limitations, and the implementation steps of the optimization method to obtain the optimal design. Each component is developed and clarified to obtain optimal MFs. The optimization technique searches the optimal solution as formulated in the objective function through manipulation of the input vector subject to the constraints in each generation of the iterative process.

4.2.1. Input vector

As a first step in FLC design, the number of MFs must be defined to provide the solution from the optimization technique. Depending on the number of MFs, the input vector Z is expressed as

$$Z_{i,j} = \left[X_{i,j}^1 X_{i,j}^2 ..... X_{i,j}^n\right] \tag{14}$$

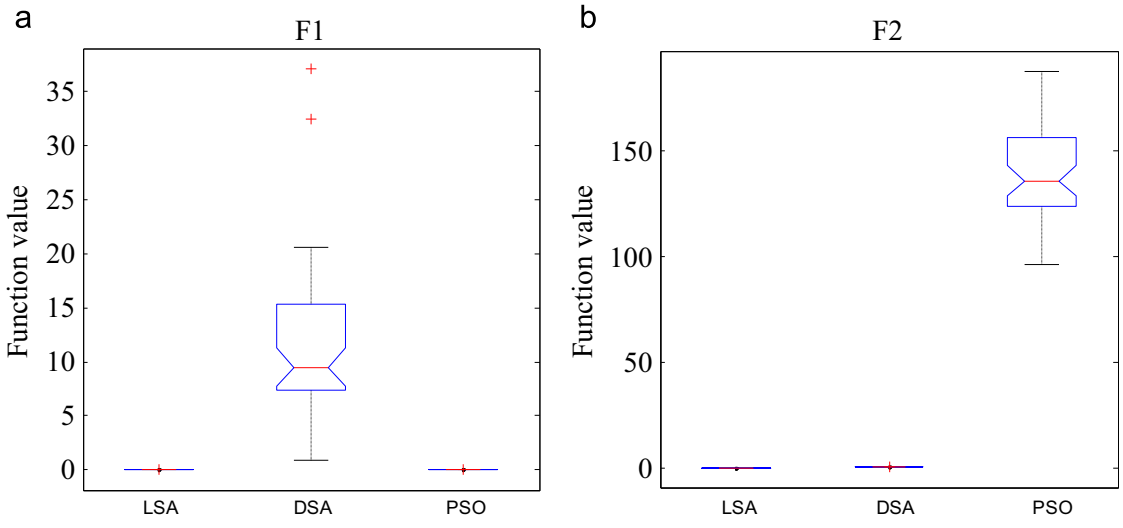

Fig. 10. Variation in global optimization results for benchmark functions in Table 3: (a) F1 and (b) F2. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

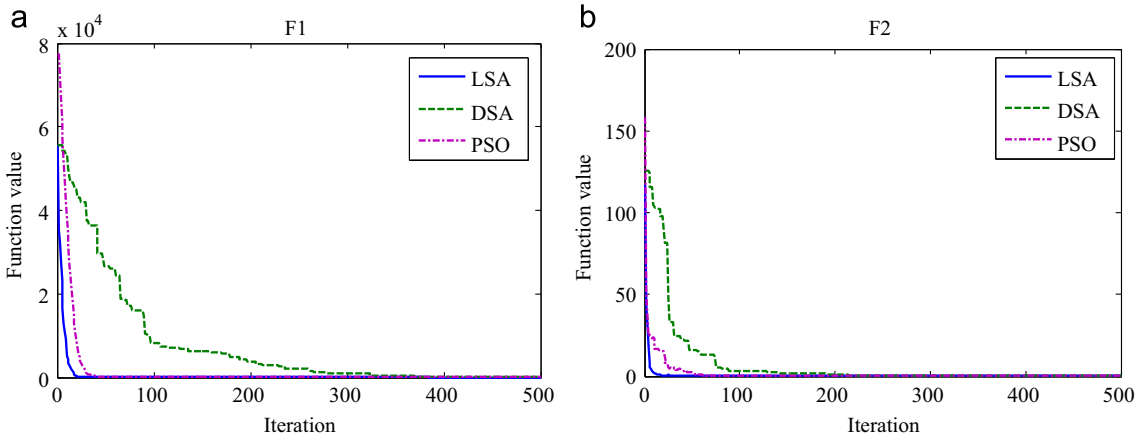

Fig. 11. Convergence characteristic curves for LSA, DSA, and PSO in solving F1 and F2.

Table 5
Unimodal and non-separable test functions.

Function ID	Name	Expression	*n*	Search space	Function minimum
F3	Schwefel 2.22	$f_3(\mathbf{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10,10]^n$	0
F4	Rosenbrock	$f_4(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	30	$[-30,30]^n$	0

where $Z_{i,j}$ represents the jth solution in the population during the ith iteration, $X_{i,j}^k$ is the kth element of $Z_{i,j}$, and n is the total number of parameters. For example, the input vector $Z_{i,j}$ should contain six parameters that indicate the boundaries of MFs to be optimized to represent the MFs in Fig. 5.

4.2.2. Objective function

An objective function is required to determine and evaluate the performance of $Z_{i,j}$ for the MFs. Thus, the objective function for finding the optimal values is formulated in such a way that $Z_{i,j}$ generates the best fuzzy control action as a crisp value according to (12) described in the defuzzification process. In the FLC design for PV inverter control, E and CE at the tth sampling step corresponding to V_d (which is the transformed inverter output voltage) indicate the goodness of the crisp value of the fuzzy control action. Therefore, the mean square error MSE (15) obtained from the reference values V_{dref} and the measured values V_d are used as the objective function.

$$\mathrm{MSE} = \frac{\sum_{i=1}^{\ell} (V_{dref} - V_d)^2}{\ell} \tag{15}$$

where V_{dref} is the reference value that is equal to (1 p.u.), V_d is the measured value, and ℓ is the number of the samples used to evaluate MSE. In the optimization process, Eq. (15) needs to be minimized.

4.2.3. Optimization constraints

The optimization algorithm must be implemented while satisfying all constraints utilized to determine the optimal values of the MF parameters. The boundaries of these parameters should not overlap. In other words, the element $X_{i,j}^k$ should be between $X_{i,j}^{k-1}$ and $X_{i,j}^{k+1}$. If the element $X_{i,j}^k$ is greater than $X_{i,j}^{k+1}$ or less than $X_{i,j}^{k-1}$, this element should be regenerated within its boundaries. Therefore, the following restriction must be fulfilled to ensure that each MF parameter is within the prescribed boundaries.

$$X_{i,j}^{K-1} < X_{i,j}^{K} < X_{i,j}^{K+1}. \tag{16}$$

Table 6
Test 2 global optimization results for benchmark functions in Table 5.

Function ID	Statistics	LSA	DSA	PSO
F3	Best	**2.21758E−07**	0.420804245	0.001456467
	Worst	0.970135549	1.939300935	**0.018902868**
	Average	0.036806544	1.006036631	**0.004923274**
	Median	**0.001069055**	0.921312778	0.004072683
	Standard deviation	0.156233023	0.357910793	**0.003334754**
F4	Best	**0.560036507**	353.5219172	14.36215241
	Worst	**201.6439055**	2347.035592	307.2355336
	Average	**64.28160301**	1108.180713	68.72292596
	Median	73.53003347	1002.19241	**67.4273481**
	Standard deviation	**43.75576111**	572.4209417	57.81076978

4.2.4. Implementation steps of LSA to obtain the optimal FLC design

The implementation starts by resetting the LSA parameters, namely, number of iterations (T), population size (N), problem dimension (D), and channel time. The initial populations for the MFs are then generated and encoded according to (14). The next step involves the evaluation of the objective function using (15). Note that a suitable running time T_r is required to populate the FLC output for the evaluation of MSE in population N. After the initial population is evaluated, the direction and position will be update using (5) and (7) respectively. After all values of $Z_{i,j}$ in the population are updated, the procedure reevaluates the objective function, and the process continues to the next iteration. This updating and objective function

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

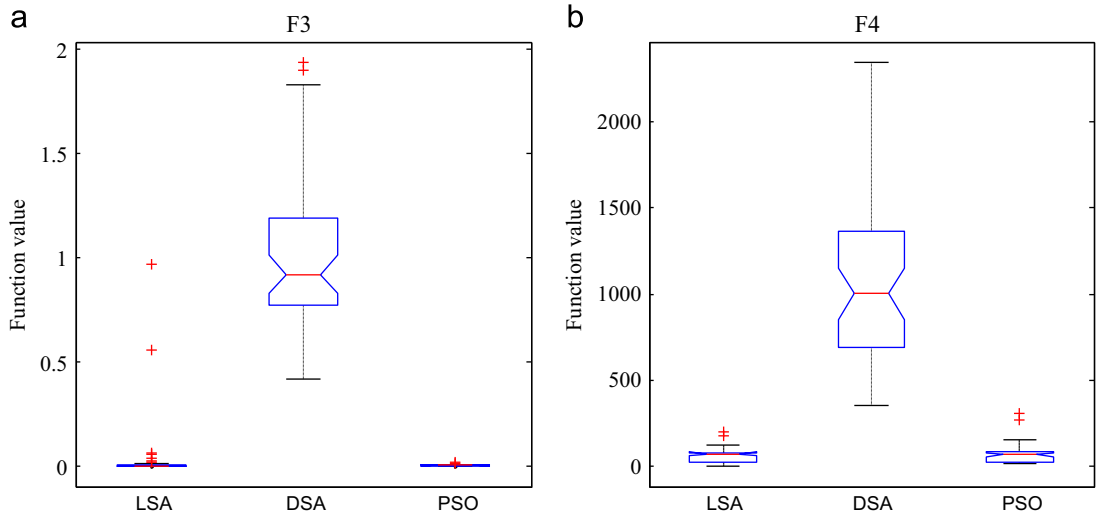

Fig. 12. Variation in global optimization results for benchmark functions in Table 5: (a) F3 and (b) F4.

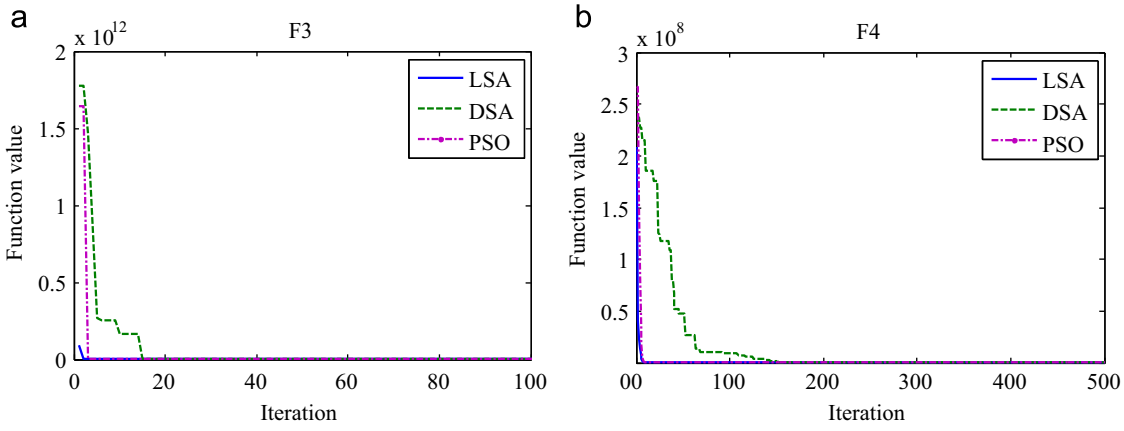

Fig. 13. Convergence characteristic curves for LSA, DSA, and PSO in solving (a) F3 and (b) F4.

Table 7
Multimodal and separable test functions.

Function ID	Name	Expression	*n*	Search space	Function minimum
F5	Foxholes	$f_5(\mathbf{x}) = \left[\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6}\right]^{-1}$	2	$[-65.53,65.53]^n$	1
F6	Branin	$f_6(\mathbf{x}) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	2	$[-5,10] \times [0,15]$	0.398

reevaluation process is repeated until the maximum iteration count is reached as explained in Fig. 7.

5. FLC design for PV inverter control using the proposed algorithm

To demonstrate the application of an optimum FLC design, a 3 kW, 240 V, 50 Hz PV inverter system is modeled in the Matlab Simulink environment (Fig. 8) to supply various types of loads continuously. As shown in Fig. 8, the three-phase outputs (V_a, V_b, and V_c) are measured and converted to V_d and V_q at each sampling time T_s=2 μs. The controller block shown in the figure contains two FLCs that correspond to V_d and $V_q$in the d–q reference frame. The controllers require *E* and *CE* to generate new V_d and V_q and to convert to V_a, V_b, and V_c. The converted signals are then utilized to generate the PWM for driving the IGBT switches of the inverter.

The MFs for each input are seven MFs defined as trapezoidal and triangular MFs are used according to the illustration in Fig. 9. Seven parameters (i.e., $X_{i,j}^1$ to $X_{i,j}^7$) are used to define the first input (*E*), whereas seven other parameters (i.e., $X_{i,j}^8$ to $X_{i,j}^{14}$) are used to define the second input (*CE*). Therefore, each controller input *Z* in the optimum FLC design contains 14 parameters. Thus, the FLC control rule for the PV inverter control system includes 49 rules (Table 1). The output is the standard output desired by the controller.

The input vector and control rules are then defined. The optimization process can be performed by evaluating the objective

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

function given in (15) using the Simulink model shown in Fig. 8 for a suitable running time of $T_r = 0.3$ s.

The optimization process based on the LSA is started by initializing the following parameters: the number of iterations (T) as 100, number of populations (N) as 20, dimension of the problem (D) as 14, and the channel time is 10. After the generation of the initial populations and the calculation the corresponding objective function for each input vector in the population, the LSA updates the population and initiates a new iteration. If the LSA reaches the maximum iteration, the FLC with the best MFs is obtained. This result indicates that the proposed method presents a systematic and easy way to design FLCs for PV inverter control systems. The following section describes the results of the problem-solving success of LSA algorithm and the optimum FLC design for the proposed PV inverter with its performance.

Table 8
Test 3 global optimization results for benchmark functions in Table 7.

Function ID	Statistics	LSA	DSA	PSO
F5	Best	**0.998003838**	**0.998003838**	**0.998003838**
	Worst	2.982105157	**0.998003838**	6.903335694
	Average	1.077446947	**0.998003838**	1.790398060
	Median	**0.998003838**	**0.998003838**	1.992030900
	Standard deviation	0.337979509	**3.36448E−16**	1.259895845
F6	Best	**0.397887358**	**0.397887358**	**0.397887358**
	Worst	**0.397887358**	**0.397887358**	**0.397887358**
	Average	**0.397887358**	**0.397887358**	**0.397887358**
	Median	**0.397887358**	**0.397887358**	**0.397887358**
	Standard deviation	**1.68224E−16**	7.79967E−12	**1.68224E−16**

6. Results and discussion

Two different tests have been carried out in this study. The first test present in Section 6.1 shows the performance of LSA algorithm in solving some of the benchmark optimization problems. The second test given in Section 6.2, shows the practical application of LSA in solving the problem of trial and error procedure in obtaining membership functions (MFs) used in conventional FLCs.

6.1. Experiment-1

The proposed LSA is tested and validated using a well-utilized set of 10 benchmark functions used by Rashedi et al. [26], Yao [35], and Venkata Rao et al. [49], while developing their optimization algorithms. However, functional characteristics such as modality, separability, and dimensionality are considerably important when testing and validating a new algorithm. The modality of a function refers to the number of vague peaks in the function surface. A function is multimodal if it has two or more vague peaks. An algorithm that encounters these peaks while searching may be trapped in one of the

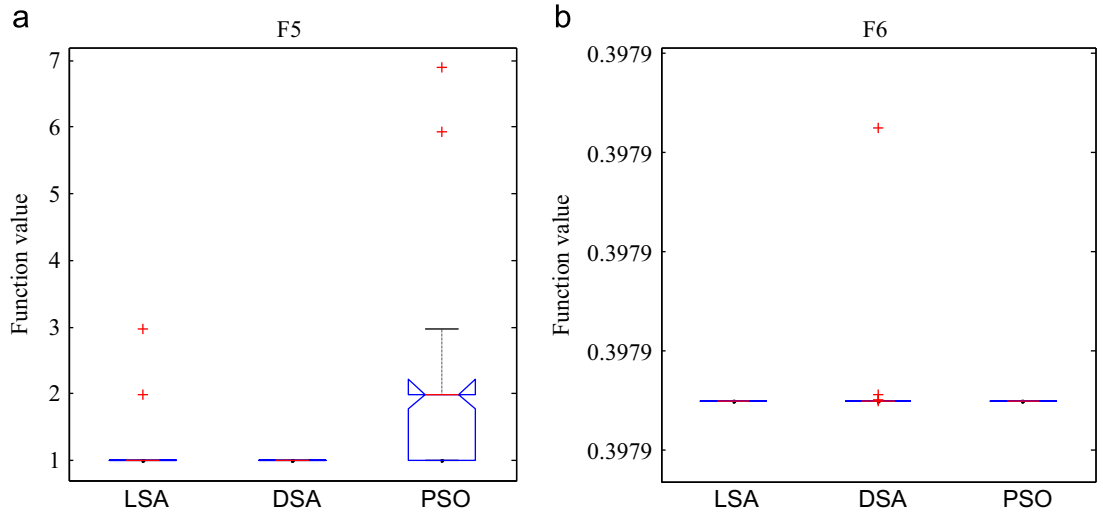

Fig. 14. Variation in global optimization results for benchmark functions in Table 7: (a) F5 and (b) F6.

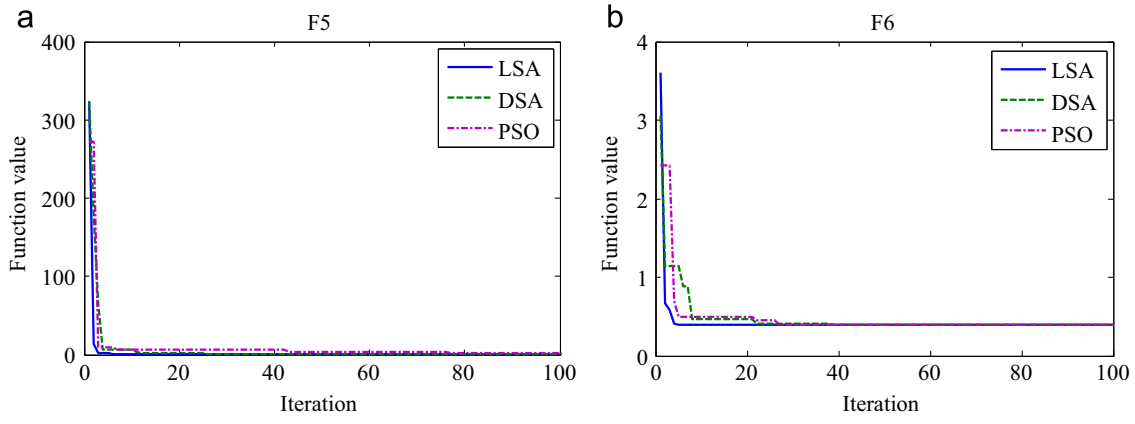

Fig. 15. Convergence characteristic curves for LSA, DSA, and PSO in solving (a) F5 and (b) F6.

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

ARTICLE IN PRESS

Table 9
Multimodal and non-separable test functions.

Function ID	Name	Expression	n	Search space	Function minimum
F7	Penalized	$f_7(\mathbf{x}) = \frac{\pi}{n}\left\{10\sin^2(\pi y_i) + \sum_{i=1}^{n-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right] + (y_n-1)^2\right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1+\frac{x_i+1}{4},\ u(x_i,a,k,m) = \begin{cases} k(x_i-a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i-a)^m & x_i < -a \end{cases}$	30	$[-50,50]^n$	0
F8	6-Hump Camel Back	$f_8(\mathbf{x}) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5,5]^n$	−1.0316285
F9	Hartman 3	$f_9(\mathbf{x}) = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j=1}^{4} a_{ij}(x_j - p_{ij})^2\right]$	4	$[0,1]^n$	−3.86
F10	Shekel 5	$f_{10}(\mathbf{x}) = -\sum_{i=1}^{5}\left[(x-a_i)(x-a_i)^T + c_i\right]^{-1}$	4	$[0,10]^n$	−10

local minima. Meanwhile, separability indicates the difficulty level of various benchmark functions. Typically, separable functions are easier to solve than non-separable functions because each variable of a function is independent of the other variables. The difficulty of a problem also increases with function dimensionality. For highly non-linear problems, dimensionality may be a significant barrier for almost all optimization algorithms. In this study, the 10 benchmark functions are categorized based on modality and separability, and four tests are conducted to evaluate the reliability and efficiency for the proposed algorithm. The LSA performance is compared with other standard heuristic optimization methods, namely, DSA, and PSO. For a fair comparison, the population size is set to 50 and the maximum iteration is set to 100, 200, and 500. The algorithm-dependent parameter settings for all algorithms are listed in Table 2.

6.1.1. Test 1

This test is conducted to evaluate the reliability and efficiency of the LSA in searching the global minimum value when it is subjected to benchmark functions with unimodal and separable characteristics. Table 3 presents the details of these functions. This test also compares the LSA with two other methods, namely, DSA, and PSO, for validation. Each benchmark function is tested 50 times. The results considering the best, worst, average, median, and standard deviation for the objective functions are shown in Table 4. The best performance for each function is boldfaced. The LSA reaches the best global minimum or near global minimum. This result is also clearly observed in the box plot (Fig. 10) constructed from the data obtained from 50 runs. Fig. 10 shows that the performance of the LSA is satisfactory for all functions because the 25th and 75th percentiles of the samples collected for the LSA fall toward the minimum solution with a narrow inter-quartile range. Meanwhile, the median of the sample for the case of PSO (indicated by a red line) is very far from the best solution value for F2.

During unimodal function optimization, the performance of the algorithm can be compared with the other methods using the convergence characteristic curves shown in Fig. 11. The LSA converges faster than the other methods and thus possesses superior convergence characteristics for this type of function optimization.

6.1.2. Test 2

To observe the performance and consistency of the LSA in solving the unimodal and non-separable functions, two benchmark functions given in Table 5 are experimented on in this test: Schwefel 2.22 (F3), and Rosenbrock (F4). These functions have the same dimensions (n=30) as those used in Test 1, but the difficulty level is higher because these functions are non-separable. The test results acquired using the LSA are compared with those obtained using the two optimization methods (Table 6). The best performance for each function is boldfaced. Table 6 shows that the LSA can find the best near-optimum solution for functions F3 and F4. To determine the consistency and overall performance of the LSA, the data obtained from 50 runs are plotted (Fig. 12). The performance of the LSA is relatively satisfactory for all the tested functions because the 25th and 75th percentiles of the samples collected for the LSA fall toward the minimum solution with a narrow interquartile range. Fig. 13 shows the convergence characteristic curves of the various optimization methods obtained while solving benchmark functions F3 and F4. Among the various algorithms, the LSA finds the best solutions.

Table 10
Test 4 global optimization results for benchmark functions in Table 9.

Function ID	Statistics	LSA	DSA	PSO
F7	Best	**2.23008E−15**	0.011463807	1.10504E−08
	Worst	4.379867485	1.468316408	**0.207337908**
	Average	0.358172550	0.291749952	**0.018659838**
	Median	0.103669020	0.170607599	**1.90335E−07**
	Standard deviation	0.743960008	0.336701290	**0.054169508**
F8	Best	**−1.031628453**	**−1.031628453**	**−1.031628453**
	Worst	**−1.031628453**	**−1.031628453**	**−1.031628453**
	Average	**−1.031628453**	**−1.031628453**	**−1.031628453**
	Median	**−1.031628453**	**−1.031628453**	**−1.031628453**
	Standard deviation	**0.000000000**	**0.000000000**	**0.000000000**
F9	Best	**−3.862782148**	**−3.862782148**	**−3.862782148**
	Worst	**−3.862782148**	**−3.862782148**	−2.810142824
	Average	**−3.862782148**	**−3.862782148**	−3.508607946
	Median	**−3.862782148**	**−3.862782148**	−3.535082985
	Standard deviation	**0.000000000**	**0.000000000**	0.307782201
F10	Best	**−10.15319968**	**−10.15319968**	**−10.15319968**
	Worst	**−2.630471668**	−10.14497223	−2.682860396
	Average	**−7.027319823**	−10.15283380	−8.653837241
	Median	−5.100772140	−10.15319541	**−10.15319968**
	Standard deviation	3.156152099	**0.001254219**	2.808248893

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

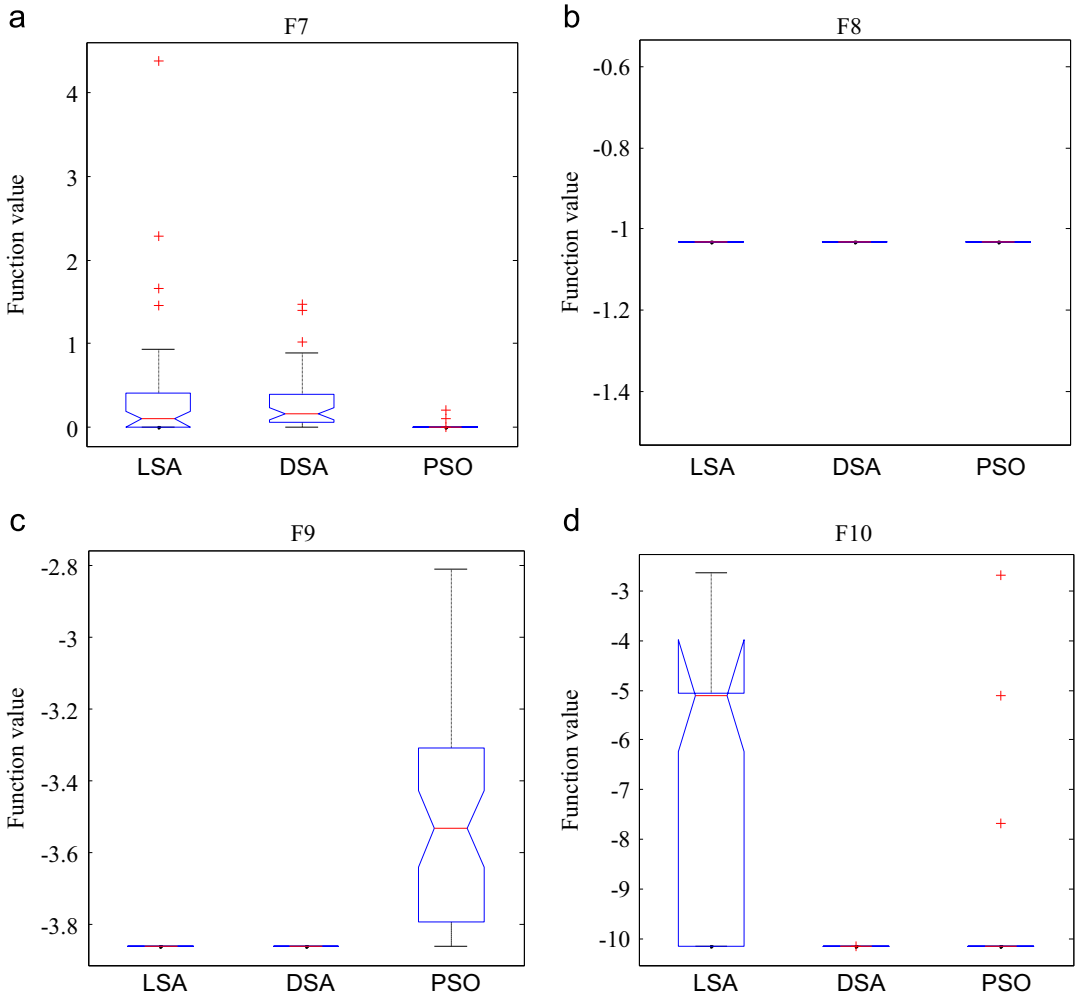

Fig. 16. Variation in global optimization results for benchmark functions in Table 9: (a) F7; (b) F8; (c) F9; and (d) F10.

6.1.3. Test 3

In this test, the difficulty level of the optimization problem is increased by using multimodal and separable functions (Table 7). Each benchmark function is tested again for 50 times. The results considering the best, worst, average, median, and standard deviation of the objective functions are shown in Table 8. The best performance for each function is boldfaced. All algorithms obtain the best global minimum or near-global minimum. However, the comparative results shown in Fig. 14 prove the capability of the LSA to escape from any local minimum. Fig. 15 shows that the convergence rate of the LSA is much faster at the initial stages than at the later stages.

6.1.4. Test 4

The exploration and exploitation abilities of the proposed LSA are also tested with four multimodal and non-separable high- and low-dimensional benchmark functions. Table 9 presents the details of these functions. Similar to the previous test analysis, this test also compares the LSA with two other methods, namely, DSA, and PSO, for validation using the same statistical indices (Table 10). The best performance for each function is boldfaced. The LSA reaches the best global minimum or near-global minimum for all tested functions. This result confirms the exploration and exploitation abilities of the proposed LSA. Moreover, the box plots shown in Fig. 16 show that the performance of the LSA is comparatively satisfactory for most of the tested functions. Nonetheless, for the Shekel 5 test function (F10), the LSA cannot find satisfactory solutions in most of the test runs because the median and the 25th and 75th percentiles of the samples collected for the LSA extend from the true solution. The convergence characteristics of the functions are depicted in Fig. 17.

6.2. Experiment-2

The PV inverter system depicted in Fig. 8 is used to evaluate the proposed LSA-based FLC (LSA-FL) optimization method and the robustness of the overall system. Fig. 18 shows the convergence characteristics of LSA-FL to find best optimal solution for the test system, along with the results obtained with DSA-based FLC (DSA-FL) and PSO-based FLC (PSO-FL). For a fair comparison, all optimization algorithms used same parameters (i.e. problem dimension, population, and iteration). Fig. 18 shows that LSA-FL converges faster than DSA-FL, and PSO-FL. Furthermore, LSA-FL generates better optimal solution compared with DSA-FL, and PSO-FL. Note that the optimum performance of LSA-FL shown in Fig. 18 is obtained when both FLCs representing V_d and V_q accomplish the MFs shown in Fig. 19. Meanwhile Figs. 20 and 21 show the optimize MF for the E and CE using DSA and PSO respectively. It should be noted from Figs. 19 to 21 that different optimization techniques gives altered membership functions for C and CE which depends on the minimum objective function value achieved at the end of optimization process. Considering the effectiveness of LSA-FL, only this controller is mainly used to

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

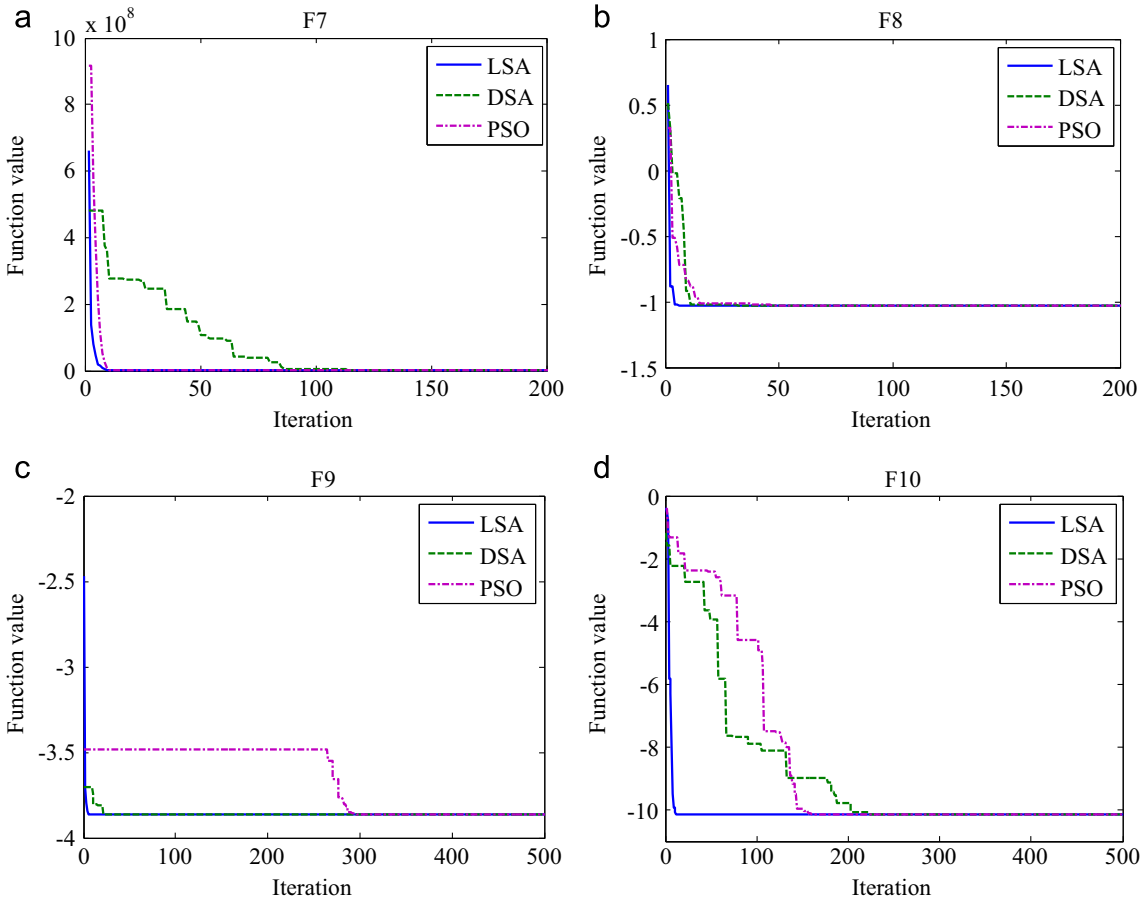

Fig. 17. Convergence characteristic curves for LSA, DSA, and PSO in solving (a) F7, (b) F8, (c) F9, and (d) F10.

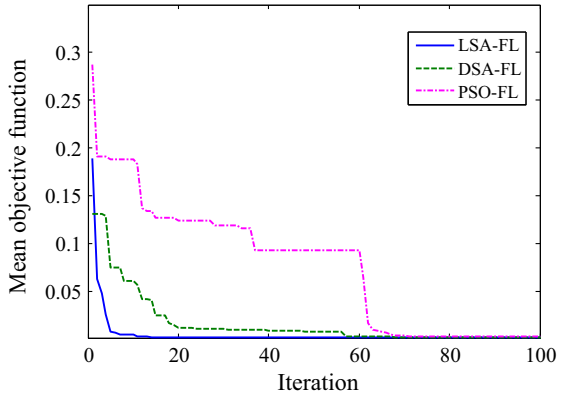

Fig. 18. Performance comparisons based on LSA-FL, DSA-FL, and PSO-FL.

evaluate the performance of the overall PV inverter system once subjected to different types of loads.

To validate the proposed inverter control strategy and to ensure that the controller can deal with different types of loads, a simulation is conducted using the simulation model shown in Fig. 8 for 0.3 s. The nonlinear rectifier load with a 100 Ω is connected from 0 s to 0.1 s. Then, the *RL* load of $R=50\,\Omega$, $L=50$ mH is connected from 0.1 s to 0.2 s. Finally, a resistive load of 50 Ω is connected from 0.2 s to 0.3 s.

Fig. 22 illustrates the output voltage waveforms with different loads. The figure shows that the voltage waveforms are not affected by changing the load type. However, short transients occur during the transition between one load to another. The voltage waveforms are sinusoidal, stable, clean, and balanced at 50 Hz and the displacement between each phase is 120°. The controller manages to precisely stabilize the amplitude at 339 V and the rms voltage of 240 V without any undesired voltage overshoots. It reveals a considerably good transient and steady-state performance of the inverter. The controller distinctly succeeds in following the exact voltage reference and quickly realizing the steady-state values.

The voltage (V_a) and current (I_a) are depicted in Fig. 23 to exhibit clearly the effect of phase voltage overshoots along with current waveform. When the nonlinear load is subjected to the inverter, the current waveform is nonsinusoidal. However, the amplitude is stable with a peak value of approximately 5.9 A. The output currents (I_a) have the same phase and frequency with the output voltage (V_a) without any problems, such as lag, lead, and flicker. The load has been changed from nonlinear load to *RL* load in 0.1 s as shown in Fig. 23. The change in load does not affect the quality of the current waveforms. The waveforms are still stable, and the controller achieves a constant peak level with approximately 6.5 A and approximately 4.5 A rms. Moreover, the waveforms are still balanced at 50 Hz and displaced by 120° between each phase. However, the load current now lags the voltage waveform by 17.45°, as depicted in Fig. 23; therefore, the power factor for this load is 95.4%. Again the load has been changed from *RL* load to *R* load as shown in Fig. 23. The phase load current waveform with *R* load shows a constant magnitude of approximately 6.8 A and a rms current of 4.8 A. This waveform is a balanced sinusoidal waveform of 50 Hz. The phase shift is also 120° between each phase. The voltage and the current have the same phase angle and therefore follow unity power factor

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

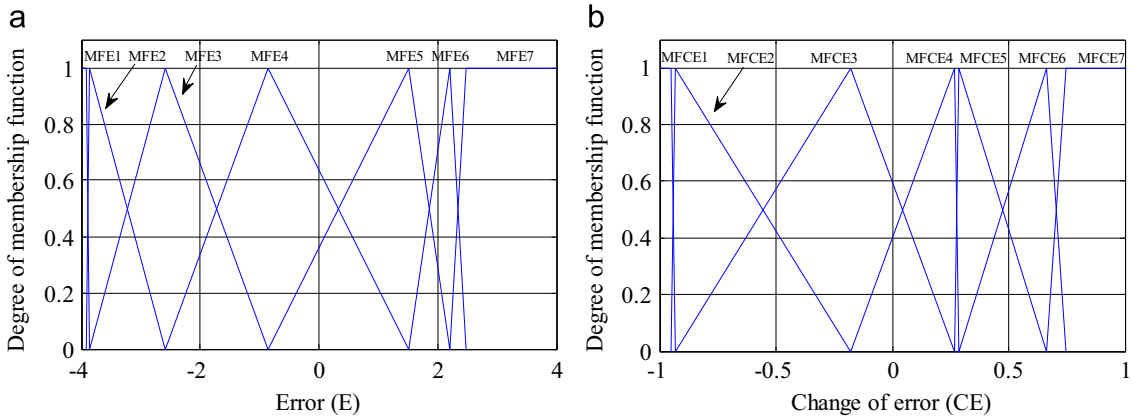

Fig. 19. Optimized membership functions of the (a) *E* and (b) *CE* using LSA.

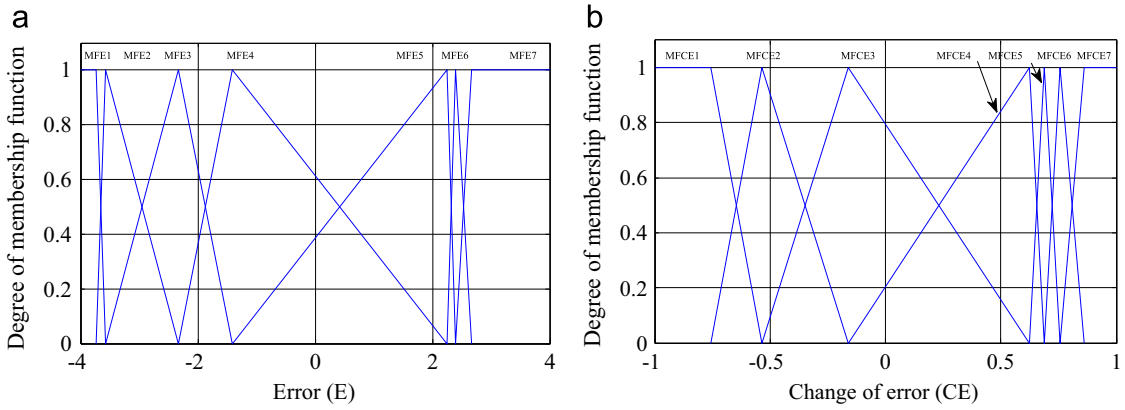

Fig. 20. Optimized membership functions of the (a) *E* and (b) *CE* using DSA.

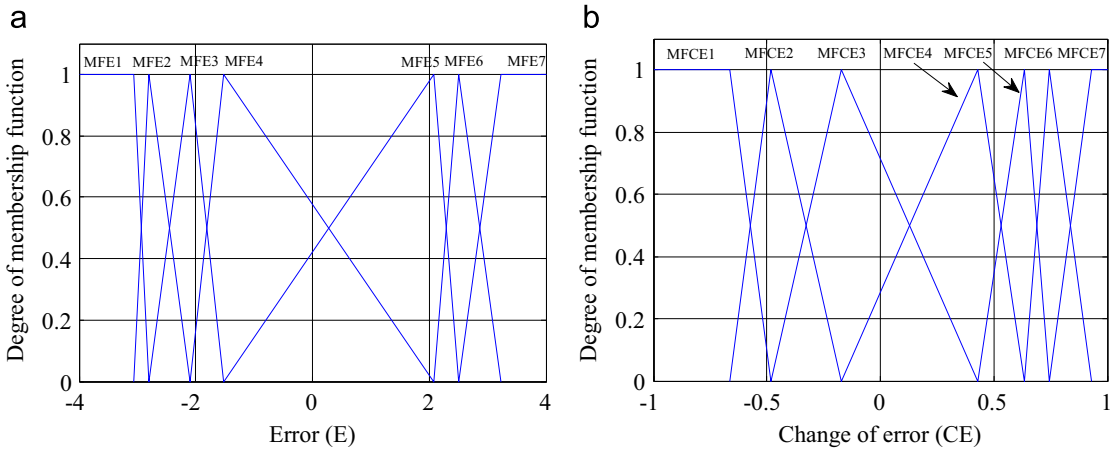

Fig. 21. Optimized membership functions of the (a) *E* and (b) *CE* using PSO.

operation as expected. Considering the phase relation of both load current and voltage waveform, it reveal in phase relationship which indicates high efficiency.

For high power capacity applications, the magnitude of the line voltage, such as V_{ab}, is commonly considered in three-phase systems. It is more than the phase voltage V_a by a value of $\sqrt{3}$. The waveform of the line voltage is illustrated in Fig. 24. The line voltage waveform (V_{ab}) does not affect system performance. Notably, the overshoots during the transient periods are minimal when line voltages are considered. Nevertheless, the controller manages to regulate the amplitude at 586 V, whereas the rms voltage is equal to 415 V as required by the system. The line output voltage acquires a sinusoidal waveform at 50 Hz, and the controller succeeds in following the exact voltage reference and quickly realizing the steady-state values.

To verify the quality on the inverter output waveforms, fast Fourier transform (FFT) is performed respecting to the total harmonic distortion (THD). The quality of the waveform is inversely proportional to the THD percentage. The THD percentage for the output waveforms should be less than 5% for the output

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

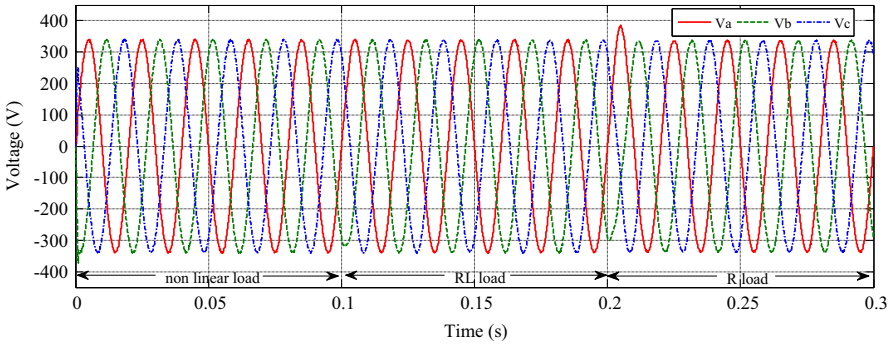

Fig. 22. Output voltage waveforms (i.e., V_a, V_b, and V_c) of the three-phase inverter with three load types.

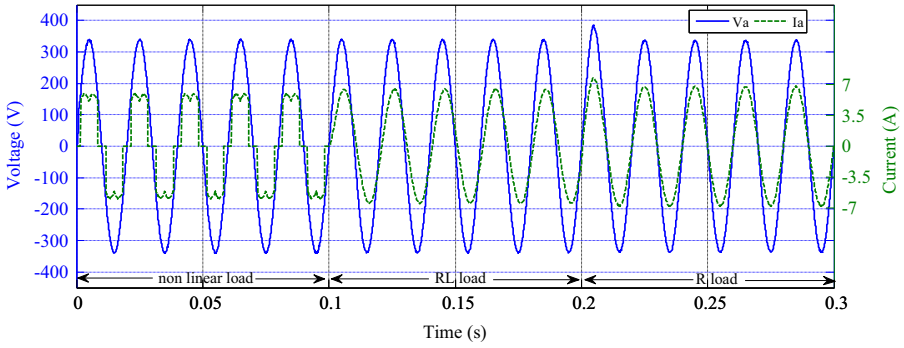

Fig. 23. Output voltage and current load of the inverter with three load types.

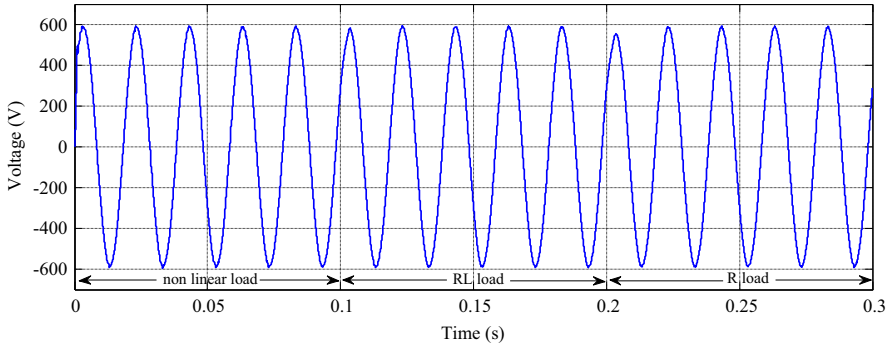

Fig. 24. Line voltage (V_{ab}) of the three-phase inverter with three load types.

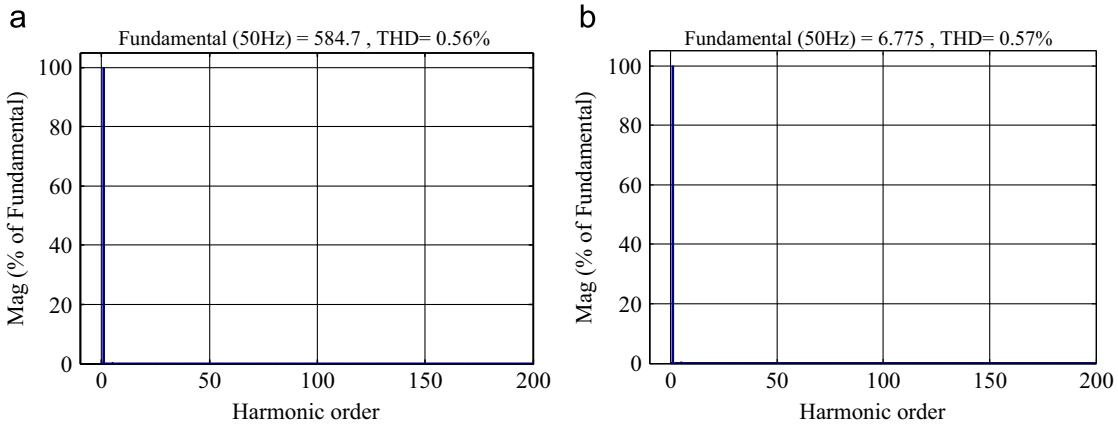

Fig. 25. THD for the three-phase inverter's phase (a) voltage and (b) current with *R* load.

freepaper.me FREE paper

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

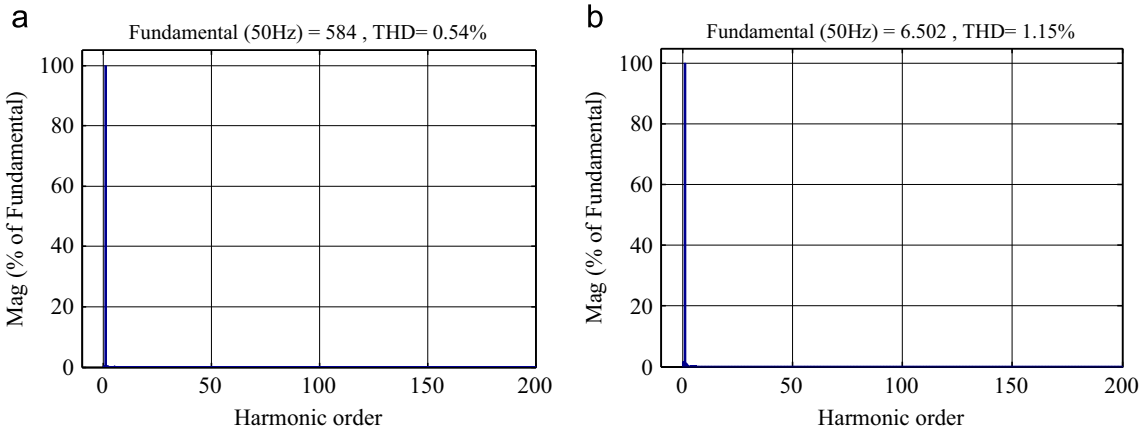

Fig. 26. THD for the three-phase inverter's phase (a) voltage and (b) current with *RL* load.

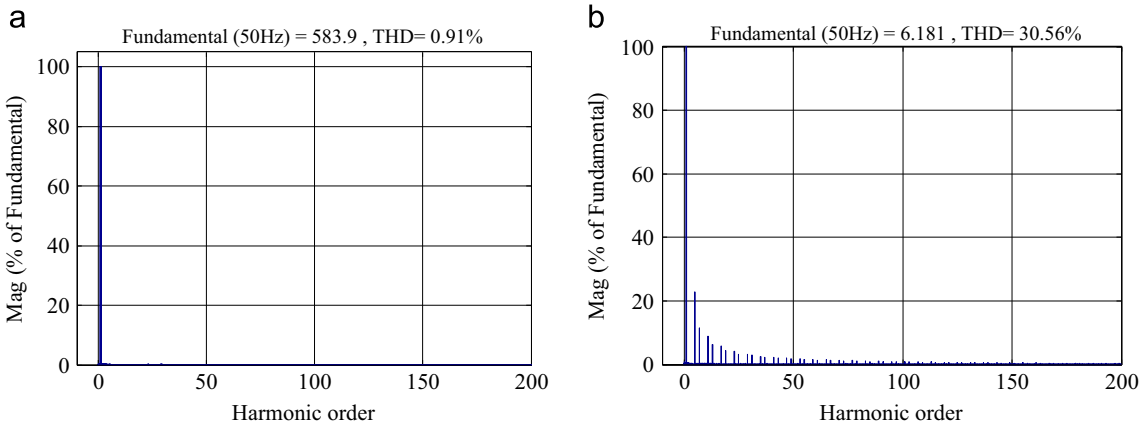

Fig. 27. THD for the three-phase inverter's phase (a) voltage and (b) current with nonlinear load.

Table 11
Mean square error.

	LSA-FL	DSA-FL	PSO-FL
R load	**0.002685594**	0.003706301	0.004859787
RL load	**0.002617753**	0.003740532	0.005186767
Nonlinear load	**0.002670313**	0.003790668	0.005780010

Table 12
Mean absolute error.

	LSA-FL	DSA-FL	PSO-FL
R load	**0.009547364**	0.016185209	0.037026325
RL load	**0.007493755**	0.019693860	0.046842405
Nonlinear load	**0.008327629**	0.020244398	0.052752357

waveforms to meet the international IEEE-929-2000 standard [50]. Figs. 25–27 show the THD percentages of the voltages and currents obtained for the three load types analyzed in this study. The controller keeps the THD of the voltage within a very small rate for all load types. This rate is much less than the 5% requirements of the IEEE-929-2000 international standard and more than 5% for the THD_i with nonlinear load. This result is attributed to the nonlinear nature of the current waveform drawn by the nonlinear load that cannot be controlled by the inverter.

To further evaluate the performance of the proposed controller compared to DSA and PSO based FLC, three indices, namely, mean square error (MSE), mean absolute error, and standard deviation (σ) for the error, were analyzed. The MSE is given by (15), and the other two indices are given by

$$\mathrm{MAE} = \frac{\sum_{i=1}^{\ell} |V_{dref} - V_d|}{\ell} \tag{17}$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{\ell} (e_d - \eta)^2}{\ell}} \tag{18}$$

where V_{dref} is the reference value that is equal to (1 p.u.), V_d is the measured value, ℓ is the number of the samples, e_d is the error between the reference value and the measured value, and η is average of the error values.

MSE is inversely proportional to the quality of the signal. A decrease in the MSE value means an increase in the quality of the signal. According to the MSE values, the LSA-FL performs better than the other controllers, as depicted in Table 11. The best performance is in boldfaced in Table 11.

Table 13
Standard deviation.

	LSA-FL	DSA-FL	PSO-FL
R load	**0.051397220**	0.060499763	0.064005389
RL load	**0.051067119**	0.060255617	0.064694449
Nonlinear load	**0.051609600**	0.060592795	0.065408192

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

freepaper.me FREE paper

Meanwhile, a small MAE value indicates that the data points are very near to the reference, whereas a large MAE value indicates that the data points are far from the values targeted. Table 12 clearly shows that the LSA-FL performs better than the other two controllers for all types of loads.

Table 13 shows the standard deviation for the LSA-FL, DSA-FL, and PSO-FL. The standard deviation show that how much the measured values are spread out from the average (mean). A low standard deviation means that most of the values are very close to the average; meanwhile a high standard deviation means that the values are spread out. According to the standard deviation index shown in Table 13, the LSA-FL again performs better than the other controllers.

7. Conclusion

This study first introduced a novel nature-inspired optimization algorithm called the LSA. It was formulated based on lightning, which originates from thunderstorm. In contrast to that of the counterparts of the LSA, the major exploration issue of the proposed algorithm was modeled using the exponential random behavior of space projectile and the concurrent formation of two leader tips at fork points using opposition theory. LSA was then proposed to automatically tune the MFs of the FLC in PV inverter. For this purpose, a suitable objective function was developed for minimizing the MSE of the output voltage at the inverter terminal. To evaluate the reliability and efficiency of the proposed algorithm, the LSA was tested using 10 benchmark functions with various characteristics and compared the performance of the LSA with DSA and PSO. The proposed LSA demonstrated satisfactory exploration, exploitation, and convergence characteristics. Furthermore, the LSA generally provides better results compared with DSA and PSO in optimizing both benchmark functions and FLC design for PV inverter.

Acknowledgment

The authors are grateful to Universiti Kebangsaan Malaysia for supporting this research financially under Grants ETP-2013-044.

References

[1] T. Khatiba, A. Mohamed, K. Sopian, Optimization of a PV/wind micro-grid for rural housing electrification using a hybrid iterative/genetic algorithm: case study of Kuala Terengganu, Malaysia, Energy Build. 47 (2012) 321–331.
[2] M.Z. Daud, A. Mohamed, M.A. Hannan, An improved control method of battery energy storage system for hourly dispatch of photovoltaic power sources, Energy Convers. Manag. 73 (2013) 256–270.
[3] F. Blaabjerg, Z. Chen, S. Kjaer, Power electronics as efficient interface in dispersed power generation systems, IEEE Trans. Power Electron. 19 (2004) 1184–1194.
[4] R. Ortega, E. Figueres, G. Garcerá, C.L. Trujillo, D. Velasco, Control techniques for reduction of the total harmonic distortion in voltage applied to a single-phase inverter with nonlinear loads: Review, Renew. Sustain. Energy Rev 16 (2012) 1754–1761.
[5] T. Ryu, Development of power conditioner using digital controls for generating solar power, OKI Tech. Rev. 76 (2009) 40–43.
[6] A.H. Mutlag, H. Shareef, A. Mohamed, M.A. Hannan, J.A. Ali, An improved fuzzy logic controller design for PV inverters utilizing differential search optimization, Int. J. Photoenergy 2014 (2014) 1–14.
[7] M. Alata, M.A. Al-Nimr, Y. Qaroush, Developing a multipurpose sun tracking system using fuzzy control, Energy Convers. Manag. 46 (2005) 1229–1245.
[8] A. Messai, A. Mellit, A. Massi Pavan, A. Guessoum, H. Mekki, FPGA-based implementation of a fuzzy controller (MPPT) for photovoltaic module, Energy Convers. Manag. 52 (2011) 2695–2704.
[9] N. Altin, I. Sefa, dSPACE based adaptive neuro-fuzzy controller of grid interactive inverter, Energy Convers. Manag. 56 (2012) 130–139.
[10] K. Premkumara, B.V. Manikandan, Adaptive neuro-fuzzy inference system based speed controller for brushless DC motor, Neurocomputing 138 (2014) 260–270.
[11] L.K. Letting, J.L. Munda, Y. Hamam, Optimization of a fuzzy logic controller for PV grid inverter control using S-function based PSO, Sol. Energy 86 (2012) 1689–1700.
[12] S. Tong, Y. Li, G. Feng, T. Li, Observer-based adaptive fuzzy backstepping dynamic surface control for a class of MIMO nonlinear systems, IEEE Trans. Syst. Man Cybern. Part B 41 (2011) 1124–1135.
[13] S. Tong, Y. Li, P. Shi, Observer-based adaptive fuzzy backstepping output feedback control of uncertain MIMO pure-feedback nonlinear systems, IEEE Trans. Fuzzy Syst. 20 (2012) 771–785.
[14] S. Tong, Y. Li, Adaptive fuzzy output feedback control of MIMO nonlinear systems with unknown dead-zone inputs, IEEE Trans. Fuzzy Syst. 21 (2013) 134–146.
[15] K. Deb, A. Kumar, Real coded genetic algorithms with simulated binary crossover: studies on multi modal and multi objective problems, Complex Syst. 9 (1995) 431–454.
[16] C. Kanzow, N. Yamashita, T. Fukushima, Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints, J. Comput. Appl. Math. 172 (2004) 375–397.
[17] P. Civicioglu, Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, Comput. Geosci. 46 (2012) 229–247.
[18] R.C. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, New York, NY, 1995, pp. 39–43.
[19] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization. Computer Engineering Department, Engineering Faculty, Erciyes University, Tec. Rep.TR-06 (1-10), October 2005.
[20] M. Dorigo, V. Maniezzo, A. Colorni, The ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. – Part B: Cybern. 26 (1996) 29–41.
[21] M. Eslami, H. Shareef, M.R. Taha, M. Khajehzadeh, Adaptive particle swarm optimization for simultaneous design of UPFC damping controllers, Int. J. Electr. Power Energy Syst. 57 (2014) 116–128.
[22] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, Appl. Soft Comput. 8 (2008) 687–697.
[23] H. Li, S. Wang, M. Ji, An Improved Chaotic Ant Colony Algorithm. Advances in Neural Networks, ISNN 2012, Springer (2012) 633–640.
[24] W. Gao, S. Liu, Improved artificial bee colony algorithm for global optimization, Inf. Process. Lett. 111 (2011) 871–882.
[25] Z.H. Zhan, J. Zhang, Y. Li, S.H. Chung, Adaptive particle swarm optimization, IEEE Trans. Syst. Man Cybern. – Part B: Cybern. 39 (2009) 1362–1381.
[26] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational search algorithm, Inf. Sci. 179 (2009) 2232–2248.
[27] Z.W. Geem, J.H. Kim, G.V. Loganathan, A new heuristic optimization algorithm: harmony search, Simulation 76 (2001) 60–70.
[28] D. Simon, Biogeography-based optimization, IEEE Trans. Evol. Comput. 12 (2008) 702–713.
[29] A. Ahrari, A.A. Atai, Grenade explosion method – a novel tool for optimization of multimodal functions, Appl. Soft Comput. 10 (2010) 1132–1140.
[30] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, New York, 1989.
[31] L.J. Fogel, A.J. Owens, M.J. Walsh, Artificial Intelligence Through Simulated Evolution, Wiley, New York, 1966.
[32] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, Glob. Optim. 11 (1997) 341–359.
[33] J.R. Koza, Genetic Programming: On the Programming of Computers by Natural Selection, MIT Press, Cambridge, MA, 1992.
[34] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31.
[35] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[36] W. Khatib, P. Fleming, The stud GA: a mini revolution?, in: A. Eiben, T. Back, M. Schoenauer, H. Schwefel (Eds.), Parallel Problem Solving from Nature, Springer, New York, 1998.
[37] J. Brest, S. Greiner, B. Boskovic, M. Mernik, V. Zumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (2006) 646–657.
[38] C. Igel, N. Hansen, S. Roth, Covariance matrix adaptation for multi-objective optimization, Evol. Comput. 15 (2007) 1–28.
[39] A.A. Dul'zon, V.V. Lopatin, M.D. Noskov, O.I. Pleshkov, Modeling the development of the stepped leader of a lightning discharge, Tech. Phys. 44 (1999) 394–398.
[40] I. Gallimberti, G. Bacchiega, A. Bondiou-Clergerie, P. Lalande, Fundamental processes in long air gap discharges, Appl. Phys. 3 (2002) 1–25.
[41] J.R. Dwyer, M.A. Uman, The physics of lightning, Phys. Rep. 534 (2014) 147–241.
[42] D. Nguyen, G. Deegan, F. D'Alessandro, Fractal nature of probabilistic model of lightning discharge, in: TENCON 2001, Proceedings of IEEE Region 10 International Conference on Electrical and Electronic Technology, IEEE 2001, pp. 814–818.
[43] A. Berkopec, Fast particles as initiators of stepped leaders in CG and IC lightnings, J. Electrostat. 70 (2012) 462–467.
[44] H.R. Tizhoosh, Opposition-Based Learning: A New Scheme for Machine Intelligence, CIMCA/IAWTIC 2005, pp. 695–701.

freepaper.me FREE paper

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083

[45] R. Nasiri, A. Radan, Pole-placement control strategy for 4 leg voltage source inverters, in: Proceedings of the IEEE Conference on Power Electronic & Drive Systems & Technologies (PEDSTC'10). IEEE 2010, pp. 74–82.
[46] G.S. Thandi, R. Zhang, K. Zhing, F.C. Lee, D. Boroyevich, Modeling, control and stability analysis of a PEBB based DC DPS, IEEE Trans. Power Deliv. 14 (1999) 497–505.
[47] C.-H. Cheng, Design of output filter for inverters using fuzzy logic, Expert Syst. Appl. 38 (2011) 8639–8647.
[48] C. Elmas, O. Deperlioglu, H.H. Sayan, Adaptive fuzzy logic controller for DC–DC converters, Expert Syst. Appl. 36 (2009) 1540–1548.
[49] R. Venkata Rao, V. Patel, An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems, Sci. Iran. 20 (2013) 710–720.
[50] IEEE Std 929-2000, Recommended Practices for Utility Interface of Photovoltaic System, The Institute of Electrical and Electronics Engineers, 2002.

Hussain Shareef received his B.Sc. with honours from IIT, Bangladesh, M.S. degree from METU, Turkey, and Ph.D. degree from UTM, Malaysia, in 1999, 2002 and 2007, respectively. He is currently a Principal Researcher at TNB Research Sdn Bhd, Malaysia. His current research interests are renewable energy technologies, power quality, artificial intelligence and power system distribution automation. He is a member of IEEE.

Ammar Hussein Mutlag received his B.Sc. and M.S. degrees from University of Technology, Iraq in 2000 and 2005, respectively. He is currently pursuing the Ph. D. degree in Electrical Engineering at Universiti Kebangsaan Malaysia. His research interest is in the application of artificial intelligence to power converters.

Azah Mohamed received her B.Sc. from University of London in 1978 and M.Sc. and Ph.D. from Universiti Malaya in 1988 and 1995, respectively. She is a Professor in the Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia. Her main research interests are power system security, power quality, artificial intelligence and renewable energy systems. She is a senior member of IEEE.

freepaper.me FREE paper

Please cite this article as: H. Shareef, et al., A novel approach for fuzzy logic PV inverter controller optimization using lightning search algorithm, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.05.083