

## On the effects of hyper-parameters adjustments to the PSO-GMPPT algorithm for a photovoltaic system under partial shading conditions

André Luiz Marques Leopoldino<sup>1</sup>, Cleiton Magalhães Freitas<sup>1</sup>, Luís Fernando Corrêa Monteiro<sup>1</sup>

<sup>1</sup>Rio de Janeiro State University, Campus Maracanã, Zip Code 20559-900, Rio de Janeiro-RJ, Brazil

### Abstract

This paper exploits the performance of the particle swarm optimization (PSO) algorithm for a photovoltaic system under partial shading condition (PSC). Essentially our main contribution consists on analyzing the hyper-parameters adjustment of the PSO algorithm to determine the minimum particle numbers, such that the assertiveness to identify the Global Maximum Power Point (GMPP) be higher than 99%. The database was obtained throughout 5760 simulations based on different test cases. From these test cases, the PSC was applied in 2880 simulations. In the previous work, it was shown the best results based on 5 particles. In this update version, it is also shown the best results for 3, 7 and 9 particles, together with a comparison among them. Furthermore, this paper also presents the simulation results to evaluate the performance of the developed algorithm under transient- and steady-state conditions.

Received on 13 May 2019; accepted on 09 October 2019; published on 22 October 2019

**Keywords:** Photovoltaic Energy Generation, Maximum Power Point Tracking, Particle Swarm Optimization

Copyright © 2019 A. L. M. Leopoldino *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi:10.4108/eai.13-7-2018.160981

### 1. Introduction

The growing concern on environmental issues caused by the lack of fossil fuels in the near future has settled the ground for the widespread of the so-called green energy sources. Among them, photovoltaic (PV) systems stands out due to the possibility of turning any household into a micro power-plant, contributing to the distributed energy concept [1]. In this context, countries such as Germany, China and Japan have taken the lead, powered either from environmental or commercial aims, on the movement of PV-energy spreading [2, 3].

From the technical point of view, it is important to highlight that, in addition to PV panels, there is a power-electronic converter controlled by maximum power point tracking (MPPT) algorithm, such that the Photovoltaic array be able to produce its maximum energy. In fact, the lack of such controllers can even turn the effective power generation impracticable. In

short, the amount of power harvested from a panel depends on the voltage across its terminals, and this relationship varies with environmental variables, such as solar irradiance and temperature. In the literature there are different proposals for MPPT algorithms. It is worthwhile noticing that methods such as Perturb and Observe (P&O), Hill Climbing, Incremental Conductance, and plenty of others based on artificial intelligence (AI) have already been extensively tested and are considered to be reliable for this purpose [4-7]. Nonetheless, these methods are prone to fail in face of partial shading condition (PSC), once they are not able to identify, correctly, the Global Maximum Power Point (GMPP).

Regarding the operation under PSC, a couple of other methods have already been proposed for MPPT [8], among which particle swarm optimization (PSO) [9]. This method belongs to the AI branch and is based on the behavior of flock of animals. Its characteristic makes it possible to search for the GMPP without being trapped into local maxima, which possible would reduce the amount of energy harvested. A brief review of the literature shows that researchers have

\*Corresponding author: Cleiton Magalhães Freitas  
Email: [cleiton.freitas@uerj.br](mailto:cleiton.freitas@uerj.br)

widely used different forms of PSO algorithms in different configurations of PV generation systems. Some authors have used PSO-MPPT for reducing steady-state oscillations [10] in a single-converter system, while others have developed algorithms for multi-converter distributed systems [11]. Besides that, it is also found in the literature PSO-MPPT applied to grid-tied systems [12]. Furthermore, some researchers have combined PSO with other techniques [13] or modified the basic concept of PSO with intention to cut off the random characteristic of the algorithm [14].

Although some of the authors previously cited have studied the influence of some parameters over the PSO-algorithm performance, there are some issues that were not deeply analyzed, such as how the number of particles chosen and the settings of the constants relates to the success rate in finding the GMPP. In views of that, this work presents a methodology to analyse the influence of the hyper-parameters (number of particles and other constants used in the algorithm) adjustment and, besides that, to determine the minimum number of particles necessary to assure assertiveness higher than 99% in identifying the GMPP. For this matter, it was considered a system comprised of a PV array represented by three series-connected solar panels, a boost converter and a 96V battery. The data from which the analysis is backed on were obtained from simulations of the system considering different number of particles — three, five, seven and nine — and different settings of the three constants used in the PSO-MPPT algorithm. Besides that, the simulation cases encompassed different irradiation conditions and different patterns of partial shading. It is important to mention that this paper complements the results presented in a previous paper [15], presenting the full set of results and discussing deeply the effects of the hyper-parameters in the performance of the PSO-MPPT. Moreover, in this update version, it is also presented simulation results on the time domain to evaluate the performance of the PSO-MPPT, set with the hyper-parameter adjustment that led to the best performance, under transient- and steady-state conditions for. Apart from these analyses, it is also in the scope of this article to present a detailed description of the PSO-MPPT algorithm.

This work is organized in nine sections as follows: Section 2 presents a review on how the PV panel is modeled and the effects of the PSC on the power vs voltage (P-V) relationship. In sequence, a fully description of the modelled circuit used to the test-case is presented in Section 3. After that, an introductory discussion on PSO algorithm is carried out in Section 4 and, in the sequence, the PSO-MPPT algorithm analyzed in this work is detailed in Section 5. Section 6 presents the methodology employed to analyze the PSO-MPPT and Section 7

presents a detailed statistical analysis of the hyper-parameters adjustment to three, five, seven and nine particle numbers. Section 8, on the other hand, presents the simulation results from the developed test case, with different PSC. One must note that PSO algorithm was carried out with the hyper-parameters adjusted to five particles, once it resulted on the best performance of the PSO algorithm as observed in Section 6. Finally, conclusions are drawn in Section 9.

## 2. PV Panels

This section presents the mathematical model used for representing the PV panels, along with a description of its behavior under different irradiance and temperature levels. Besides that, it is also explained the effects of PSC on the P-V curve.

### 2.1. PV Panel Modeling

The single diode model approach [16] was used to represent the PV panel. It corresponds to a current source,  $I_{ph}$ , paralleled with direct biased diode and a shunt resistor,  $R_{sh}$ , as it is presented in Figure 1. Besides that, a series resistor,  $R_s$ , is also inserted in the circuit model. As the shunt resistance tends to assume high values [17], the current flowing out of the PV panel may be expressed by:

$$I_{pv} = I_{ph} - I_d \quad (1)$$

where  $I_d$  is the current flowing through the diode.

Basically, the current source,  $I_{ph}$ , models the panel response in function of the solar irradiance ( $G$ ) and, moreover, of the environment temperature ( $T$ ). Based on those parameters,  $I_{ph}$  is given by:

$$I_{ph} = \frac{G}{G_{ref}} I_{sc,ref} + C_T (T - T_{ref}) \quad (2)$$

where  $G$  and  $T$  are, respectively, the irradiance in  $W/m^2$  and temperature in  $K$  which the panel is submitted, whereas  $G_{ref}$  and  $T_{ref}$  are the reference values of these variables for which the short-circuit current  $I_{sc,ref}$  was measured. Regarding to the parameters  $G_{ref}$ ,  $T_{ref}$  and  $I_{sc,ref}$ , they, generally, are empirically determined by

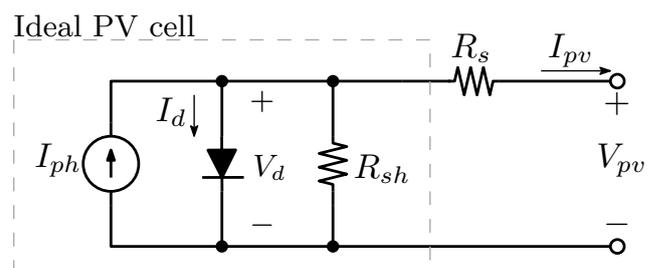


Figure 1. Single-diode equivalent circuit of a PV cell.

the manufacturer. As for the constant  $C_T$ , it is simply a temperature coefficient that accounts the effect of this variable into the short circuit current.

The diode current, on the other hand, can be expressed by:

$$I_d = I_o \left( e^{\frac{V_d}{aV_t}} - 1 \right) \quad (3)$$

where  $I_o$  is the saturation current of the cell,  $V_t$  is the thermal voltage of the PV and  $a$  is a factor which depends on the doping of the silicon used in the panel [18]. Different from  $I_o$  and  $a$ , which are constants provided by the manufacturer in the data-sheet,  $V_t$  is computed by means of the following formula:

$$V_t = \frac{kT}{q} \quad (4)$$

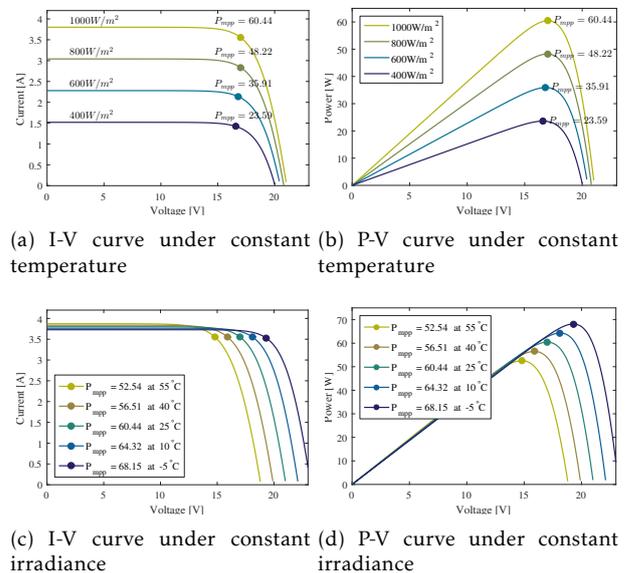
where  $k$  is the Boltzmann constant,  $1.38 \times 10^{-23} \text{J/K}$  and  $q$  is the electron charge,  $1.6 \times 10^{-19} \text{C}$ . The main data of the panel considered throughout this work is summarized in Table 1.

Considering the PV panel, for which the technical parameters are presented in Table 1, one may find out the I-V and P-V characteristic curves displayed in Figure 2. The highlighted dots on Figure 2 indicate the corresponding maximum power to the I-V curve and P-V curve under constant temperature or constant irradiance. It is possible to notice from Figures 2(b) and 2(d) how the MPP changes in function to the different levels of irradiance and temperature.

## 2.2. Characteristics of the PV Panel under PSC

The PSC occurs when a PV panel or an array of panels is submitted to non-homogeneous distribution of irradiance. It means that, due to some external factors such as clouds, leaves or even birds and others animals covering part of the panels, each panel is submitted to different equivalent irradiance levels. Such situations are known as PSC. When in PSC, either cells or even full panels turn into loads for those associated to them, which requires the use of bypass diodes for enhancing

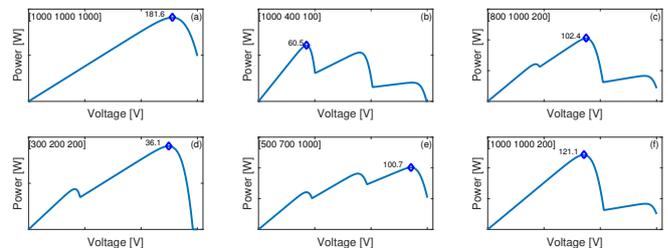
power generation [19]. Nonetheless, the result of having bypassed panels under PSC is the presence of local and global maximum power-points in the P-V curve, as illustrated in Figure 3. It must be pointed out that the position of the MPPs and quantity of them depends on factors such as number of cells/panels shaded and unshaded, irradiance and temperature. It is also important to realize that the peak value of each maximum not necessarily matches the others and that there is no straightforward rule to identify the GMPP. Hence, MPPT algorithms can be confined into local maxima rather than the global, such that, in this case, the MPPT algorithm must be able to identify the GMPP.



**Figure 2.** I-V and P-V curves of a PV panel. (a) and (b) presents the I-V and P-V curves for different values of irradiance keeping temperature constant at  $25^\circ\text{C}$ . (c) and (d) presents the I-V and P-V curves for different values of temperature keeping irradiance level at  $1000 \text{W/m}^2$ .

**Table 1.** Parameters of the used panel.

Parameter	Symbol	Value
Referential irradiance	$G_{ref}$	$1000 \text{W/m}^2$
Referential temperature	$T_{ref}$	$298.15 \text{K} (25^\circ\text{C})$
Short-circuit current	$I_{sh,ref}$	$3.8 \text{A}$
Saturation current	$I_o$	$2.16 \times 10^{-8} \text{A}$
Diode coefficient	$a$	$1.12$
Thermal constant	$C_T$	$0.0024 \text{A/K}$
Rated power	-	$60.53 \text{W}$
Rated voltage	-	$17.04 \text{V}$
Number of cells	-	$36$



**Figure 3.** P-V curve shapes for an array of three series-connected panels under PSC. The vector  $[G_1 \ G_2 \ G_3]$ , placed on each upper left corner, represents the irradiance distribution in the panels for that case and the diamond-shaped mark spot the GMPP.

### 3. Proposed Scenario and Methodology

Figure 4 presents the scenario considered for evaluation of the PSO-based MPPT. Basically, an array of three series-connected PV panels supply a bank of batteries through a boost converter. It is important to notice that an input capacitor,  $C_{in}$ , is paralleled with the PV array just on the input of the converter. This capacitor plays an important role in the circuit because it is the storage element responsible to sustain the voltage across the PV array,  $v_{pv}$ . It is also important to notice that it is possible to change  $v_{pv}$  throughout the duty cycle,  $d$ , of the converter. One last point must be addressed about the converter which drives the panel: its operational characteristic does not exactly matches the characteristic of the conventional boost converter. First because the terminal which the higher voltage is obtained is kept constant (the 96V battery does that) and second because the low-voltage terminal is fed by a current source (the PV panel). The main outcome of these changes is that unitary duty cycle is not a threat for the system. In fact, when the duty cycle tends to 1,  $v_{pv}$  tends to zero. On the other hand, when the duty cycle tends to zero,  $v_{pv}$  tends to  $V_{dc}$ .

Still on the Figure 4, there is control block labelled as *Control Algorithms*, which contains the PSO-MPPT algorithm. As it can be seen, both voltage  $v_{pv}$  and current  $i_{pv}$  of the panels are the inputs of the control algorithm and, through them, the produced power by the PV array is calculated in real time. The output of this block is the duty cycle  $d$ , which is used to drive the converter. Table 2 presents the parameters used during this paper. One may note that  $C_{in}$  is 10 times more smaller than the filter capacitor ( $C_f$ ), once, in this case, such capacitor at the input of the dc-dc converter is only necessary to avoid sudden voltage variations that might compromise the PSO-MPPT performance. Furthermore, the passive elements  $L_f$  and  $C_f$  were chosen for a 10kHz switching frequency, with the dc-dc converter under continuous mode of operation for a duty-cycle no lower than 0.4, under steady-state condition.

It is worthwhile noticing that, as the considered system comprises three PV panels externally bypassed by three diodes, such as in Figure Figure 4, there may be up to three MPPs. In this regard, if one of the panels

**Table 2.** Reference for parameters used throughout simulations.

Parameter	Symbol	Value
Input capacitor	$C_{in}$	30nF
Filter Inductor	$L_f$	0.5mH
Filter capacitor	$C_f$	50 $\mu$ F
Battery Voltage	$V_{dc}$	96V
Switching Frequency	$f$	10kHz

is under PSC, is equivalent radiance level will differ from the others and the system will present two MPPs. Moreover, if each of the panels presents a different irradiance level, the system will present three MPPs.

The presented system was simulated in PSIM considering the switches (transistor and diodes) and the battery as ideal devices. Besides that, each PV panel was modeled as described in the section 2.1 and the MPPT was set to run in a realistic update frequency (10kHz). In a general way, the model represents the main characteristic of the real-world system, and only a single point is not covered: the influence of measurement noise. Nevertheless, despite not representing the measurement noise in the simulation, the algorithm was designed to deal with it, as it will be shown in section 5. In a nutshell, the presented algorithm is ready for a real-world implementation.

### 4. PSO

Basically, the PSO is an AI-based method, similar to genetic algorithms, based on the behavior of flock of birds and others animals [20]. It was observed that the individuals of such groups take advantage of the so call group intelligence to achieve their collective goal, as finding shelter or food for instance. Therefore, if the objective is to find food, rather than concentrating all the individuals together, the flock is spread over a large area and, as the members communicate with each other, all of them converge to the place which is supposed to have more food available. In the same way, a group of individuals, in this case called particles, is part and parcel of the PSO method. Indeed, a predefined number of particles are spread all over the domain of the problem, and they are programmed to gather information and to interact with each other to find the solution, which, in general, is the point in the domain function which corresponds to the maximum or minimum of the cost function. As the cost function may present local minimum, finding the global inflection point requires well setting of the PSO algorithm, besides the widespread placement of particles.

Thus, one may note that the PSO is implemented through an iterative algorithm. Each particle is initially placed at different hyperspaces and, after each iteration (season), they move around, with an update speed factor, towards the solution of the problem [21]. It is important to notice that all of the hyperspaces are mapped on the domain of the problem and, as it has already been stated, the problem itself is to find the minimum point of a user-defined cost function. The velocity  $u_i$  with which a certain particle moves is given

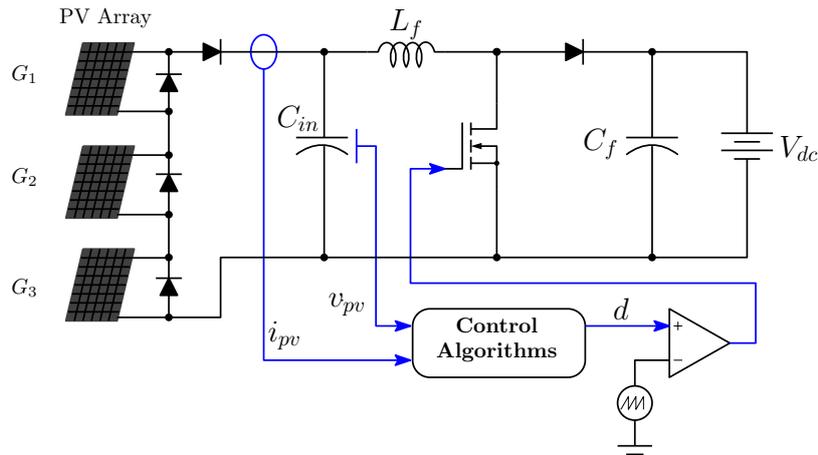


Figure 4. Simplified electrical scheme of the simulated circuit.

by:

$$\begin{aligned}
 u_i[k] = & w u_i[k-1] \\
 & + \underbrace{C_1 r_1 (pbest_i - x[k-1])}_{\text{cognitive component}} \\
 & + \underbrace{C_2 r_2 (gbest - x[k-1])}_{\text{social component}} \quad (5)
 \end{aligned}$$

The number of the particles is represented by the  $i$  parameter,  $k$  is the number of the iteration,  $w$  is the inertia weight,  $C_1$  and  $C_2$  are positive constants,  $r_1$  and  $r_2$  are random numbers ranging from 0 to 1, and  $x$  is the position of the particle. Besides that, the variables  $pbest$  and  $gbest$  correspond to the best positions (positions which returned the best results) achieved, respectively, by this particle and by the whole group, accounting all of the previous iterations.

As it can be seen in (5), the velocity of a particle changes with two main components named cognitive and social. The former is intended to draw upon the own experience of each particle, whereas the later focus on the group acquired knowledge. It is worth noticing that the chosen values of the constants  $C_1$  and  $C_2$  are straightforwardly linked with the dynamic of the algorithm and because of that, these values must be adequately tuned according to the objective. As for the random variables  $r_1$  and  $r_2$ , they play an important role in the searching process, avoiding the particles to rapidly settle on an unchanging direction of movement [20]. Other point noteworthy is the role of inertia weight  $w$ . The greater this coefficient is, the wider the domain is explored. In other words, higher values of  $w$  promotes better the searching for global minimum of the cost function [21]. Nonetheless, there are a couple of issues that can arise from choosing huge values of  $w$ . Firstly, as the  $w$  grows the speed of convergence of the particles is slowed and a tread-off

between the quality and the speed of the searching is raised. Secondly, and probably more important, huge values of  $w$  can make the particles move to the edge of the domain, causing exactly the contrary of the desired effect. Consequently, the value of  $w$  must be great enough for spreading the search all over the domain, yet not too big because it would make them concentrate on the edge of the domain. Finally, the position of the particle  $i$  in the  $k^{th}$  iteration is update as follows:

$$x_i[k] = x_i[k-1] + u_i[k] \quad (6)$$

After that the current season is finished and a new season starts, if the stop criteria were not achieved. Further information of how the PSO algorithm was implanted was described in sequence.

## 5. PSO-MPPT

As the voltage across the panels can be controlled by means of changing the duty cycle,  $d$ , this parameter was chosen to represent the domain of the problem. Thus, the position of each particle, this last being referred as  $q_i$  in some graphs, represents a duty cycle and it was updated as follows:

$$d_i[k] = d_i[k-1] + u_i[k-1] \quad (7)$$

It should be borne in mind that  $d_i[k]$  must be bounded within the interval  $[0, 1]$ , otherwise the PSO algorithm may command a searching outside of the domain problem. In this work, it was decided to simply force  $d_i = 0$  or  $d_i = 1$  whenever they reach values lower than zero or greater than one, respectively. Once the domain

problem was defined, (5) can be rewritten as:

$$u_i[k] = w u_i[k-1] + \underbrace{C_1 r_1 (d_{best,i} - d[k-1])}_{\text{cognitive component}} + \underbrace{C_2 r_2 (d_{best,g} - d[k-1])}_{\text{social component}} \quad (8)$$

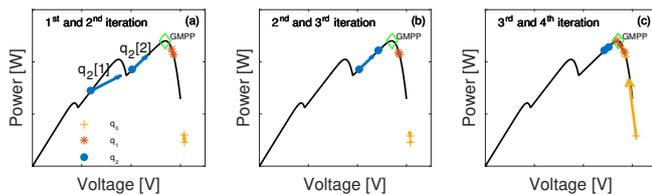
In this case,  $u$  is the change rate of the duty cycle, which consists on the particle speed, with  $d_{best,i}$  and  $d_{best,g}$  corresponding, respectively, to the  $pbest$  and  $gbest$  variables that were described in (5). Although they are not used in (8) it is important to point that the values of associated power to  $d_{best,i}$  and  $d_{best,g}$  shall be stored, respectively, in  $P_{max,i}$  and  $P_{max,g}$ . Unfortunately (8) has no bounds and must not be used in this form, otherwise it could threaten the performance of the algorithm. Among different methods for limiting the rate of changing of the duty cycle, it was chosen the trigonometric approach used in [22]. Thus, (8) was rewritten as:

$$u_i[k] = \frac{2}{\pi} \tan^{-1} \left( w u_i[k-1] + C_1 r_1 (d_{best,i} - d_i[k-1]) + C_2 r_2 (d_{best,g} - d_i[k-1]) \right) \quad (9)$$

As the image of the inverse tangent corresponds to  $[-\pi/2, \pi/2]$  and it only returns  $\pm\pi/2$  in case the argument reaches  $\pm\infty$ , this formula bounds the velocity within the interval  $(-1, 1)$ .

When it comes to the cost function, the approach presented in (4) was changed according to the current objective so as to achieve the MPP. Hence, the algorithm was arranged to search for a maximum instead of a minimum of a function, and its output corresponds to the produced power by the PV arrays.

Once explained how the position and velocity of each particle are computed, it is time to present the PSO-MPPT algorithm. Firstly, all particles have to be initialized, and this takes an important role in the



**Figure 5.** Particle swarm convergence for three particles,  $q_0$ ,  $q_1$  and  $q_2$ . The arrows sizes are proportional to the particles speeds and the graphs show the paths from (a) 1st to 2<sup>nd</sup>, (b) 2<sup>nd</sup> to 3<sup>rd</sup>, and (c) 3<sup>rd</sup> to 4<sup>th</sup> iterations.

process. For proper convergence it is necessary not only a certain minimum number of particles, but also that these are properly spread throughout the domain of the problem. To better understand this process, Figure 5 exemplifies the desired behavior for a case with three particles. In the leftmost square is presented the particles in their initial position. Note that rather than displaying actually the position (duty cycle) it was chosen to spot the point into the P-V curve associated to it. After a couple of turns, the particles move to the positions shown on the central graph and latter on they converge to the GMPP, as shown in the graph on the right. In order to have a faster and more accurate convergence process it was assigned different ranges for the initialization of each particle. It means that in case of having  $n$  particles, their initial positions might be a random number within specific intervals, as follows:

$$\begin{aligned} d_1[0] &= \text{rand} \left( \left( 0, \frac{1}{n} \right) \right) \\ d_2[0] &= \text{rand} \left( \left( \frac{1}{n}, \frac{2}{n} \right) \right) \\ &\vdots \\ d_n[0] &= \text{rand} \left( \left( \frac{n-1}{n}, 1 \right) \right) \end{aligned} \quad (10)$$

In this case,  $\text{rand}(\cdot)$  is a function which returns a random number within the interval given. This approach guarantees the widespread placement of the particles in the first turn (season). It was also chosen  $u_i[0] = 0$  as initial velocity for each particle.

After setting the initial states of the particles, the iterative process goes off. Every cycle  $k$ , the positions  $d_i[k]$  of all  $n$  particles are updated according to equation (7) and, then, the converter is driven with each value in sequence to compute the generated power  $P_i[k]$ , from  $v_{pv}$  and  $i_{pv}$ , associated to each particle. Notice that whenever the duty cycle is changed the circuit undergoes a transient. That is why there is a delay of  $4.5ms$  to avoid a wrong interpretation of the current power produced by the PV string. In addition to that, instead of using the instantaneous power, it was chosen to be used the average power computed over 50 samples (covering a period of  $0.5ms$ ) to guarantee that any oscillation or noise in the signals do not compromise the proper work of the algorithm. Once the sampling frequency of the algorithm corresponds to  $20kHz$ , the average component of  $P_i[k]$  is calculated with 10 samples. In case the  $P_i[k]$  surpass the maximum power,  $P_{max,i}$ , achieved by the particle so far,  $d_i[k]$  is assigned to  $d_{best,i}$ , and  $P_i[k]$  to  $P_{max,i}$ . Before finishing the iteration, after all particles have been tested,  $P_{max,g}$  and  $d_{best,g}$  are updated, case any  $P_{max,i}$  surpass its previous value, and the velocities  $u_i[k]$  are computed. Attention has to be raised to the fact that even after converging to the MPP, the particles still continue changing position

due to the random characteristic of the process and it can cause oscillations on the power generated. Thus, it was chosen to halt the algorithm and drive the converter with  $d_{best,g}$ , whenever the particles crowd each other in a small neighborhood. To infer if the particles are next to each other is used the standard deviation of the average point of  $P_i[k]$ , given by:

$$\sigma[k] = \sqrt{\frac{1}{n} \sum_{i=1}^n (P_i[k] - \mu[k])^2} \quad (11)$$

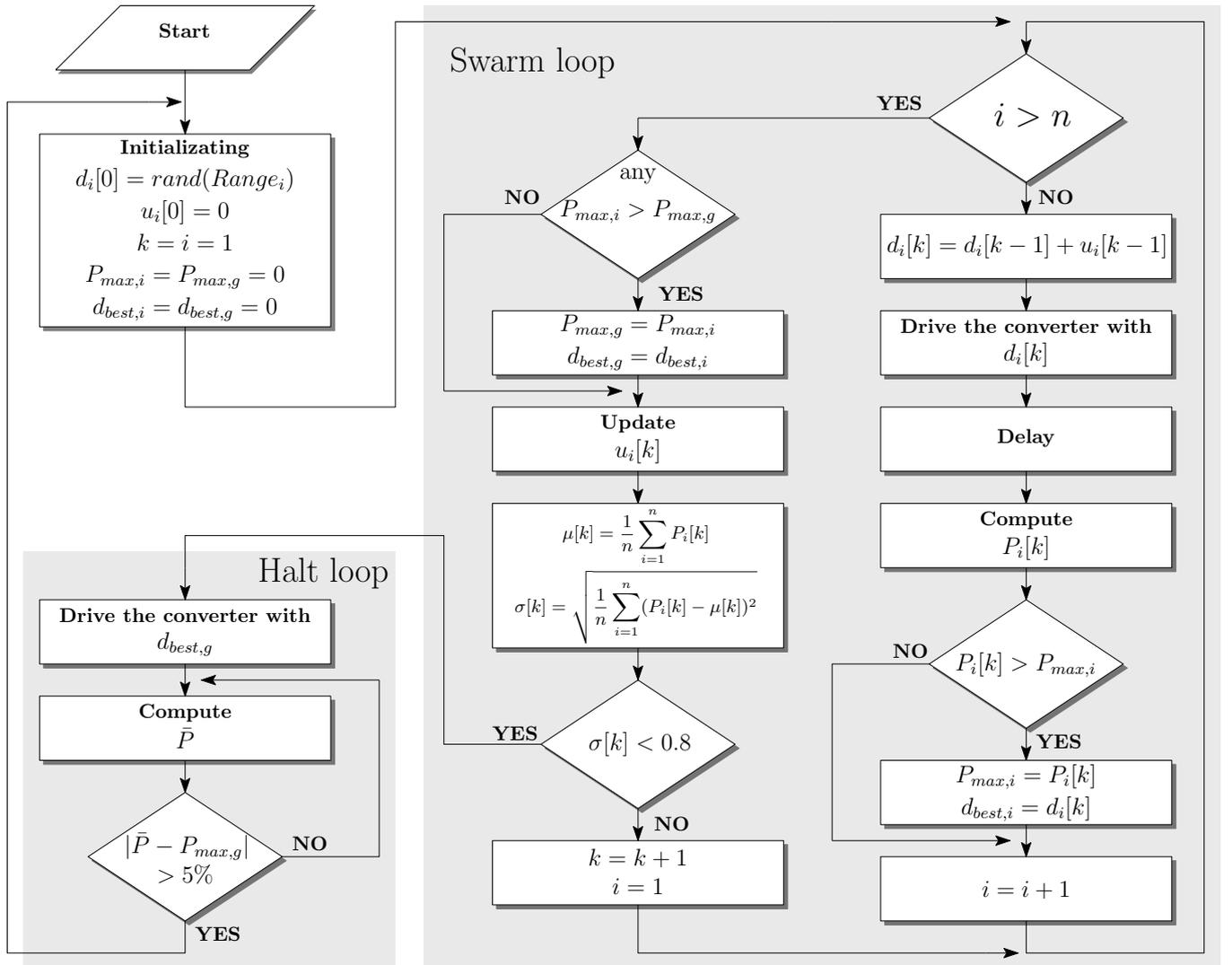
where

$$\mu[k] = \frac{1}{n} \sum_{i=1}^n P_i[k] \quad (12)$$

is the average power of all particles after the  $k^{th}$  iteration. The halt loop is commanded to arise whenever  $\sigma[k] < 0.8$  is satisfied. The flowchart in Figure 6 summarizes the algorithm. Notice that once halt, the searching process only restarts if the average component of the active power, calculated in the current iteration, presents an oscillation higher than 5%, in comparison to the calculated in the previous one. This approach prevents that measurement noise spark a new search.

## 6. Methodology Applied to Analyze the Hyper-parameters Influence on the Performance of the System

This section aims at analyzing the performance of the PSO-MPPT under different setting conditions, it means,



**Figure 6.** Particle swarm algorithm flowchart for  $n$  particles. The *swarm loop* represents the process of going from particle to particle testing the power response. The *halt loop* goes off whenever the particles converge to GMPP. It was decided not to show the limiter of  $d_i$  in this diagram for the sake of space.

different number of particles, values of the constants  $w$ ,  $C_1$  and  $C_2$ . Firstly, it was defined a set of values of the parameters for which the PSO algorithm was to be analyzed. The number of particles, for instance, was set for three, five, seven and nine. The constants  $C_1$  and  $C_2$ , on the other hand, were varied from 1 to 2 with steps of 0.2 and, the inertia constant,  $w$ , was varied from 0.2 to 1.0 with the same step size (0.2), totaling 720 different combinations of setting points. Table 3 summarizes the ranges in which each variable of the simulations was varied.

Considering this space, a set of eight simulations were carried out for each combination to assess the performance of the algorithm under different conditions. In half of them the PSC was considered, each one with a different pattern, and the other half full coverage of the sun, with different levels of irradiance. The temperature was held constant at 25°C for all the simulations. In summary, it was carried out 5760 simulations, 2880 of which the PSC was considered.

### 7. Analysis of the Results

For the sake of classifying the results, they were binarily classified into group A, for those which reached accuracy higher than 99% (virtually 100% accuracy), and group B, for all the other results. In this context, accuracy represents the per unit value of the power produced by the converter taking as reference the theoretical MPP for each case. The global results of simulations and classification are shown in histogram in Figure 7. As expected, 100% of accuracy was obtained when the PSC was not considered. Meanwhile, under PSC, the tests unveiled that only 87.67% (2525) were classified into the group A. Here it is important do state that among those nearly 12% which were classified into the group B, some ended up in a local maxima and others did not converged.

In sequence, to compare the algorithm performance with the corresponding set of parameters it was defined

Table 3. PSO search range parameters.

Parameter	Range	Step	Unit
$w$	(0.2, 1)	0.2	–
$C_1$	(1, 2)	0.2	–
$C_2$	(1, 2)	0.2	–
$n$	(3, 9)	2	–
$G_1, G_2, G_3$	(400, 1000)	200	$Wm^{-2}$
$T_{env}$	25	0	°C

the success rate,  $SR$ , of the MPPT as follows:

$$SR(i, j, m, l) = \frac{|A|}{|A| + |B|} \Big|_{C_1(i), C_2(j), w(m), n(l)} \begin{cases} i, j \in \mathbb{N}^* \mid i, j < 7 \\ m \in \mathbb{N}^* \mid m < 6 \\ l \in \mathbb{N}^* \mid l < 5 \end{cases} \quad (13)$$

In this case,  $|A|$  and  $|B|$  represents, respectively, the number of cases classified in groups A and B for a specific combination of parameters  $n$ ,  $C_1$ ,  $C_2$  and  $w$ . In short,  $SR(i, j, m, l)$  informs the percentage of cases which were classified in the group A for each combination of parameters.

Since the swarm size  $n$  is the principal hyper-parameter in the algorithm, we partitioned the  $SR$  using this metric. For each number of particles, we tested 720 scenarios and the success rate  $SR$  for them is shown in Figure 8. In these graphs, the colors blue and yellow represent the extremities of the results: the lowest and the highest levels of  $SR$ , respectively. Thus, one must note from Figure 8 (b) that there are 13 combinations of  $C_1$  and  $C_2$  regarding the case with  $n = 5$  which the success rate achieved 100%, indicating that all these tests (260) converged to the GMPP. Notice that the inertia constant varies among all these tests and it is going to be analyzed in the sequence.

Looking into the charts of the Figure 8 is possible to notice that the higher values of success rate occurred for five particles in a region bounded by  $C_1 < 1.8$  and  $C_2 > 1.4$ . On the other hand, the worst results came out of the case with nine particles, where an average of roughly 85% success rate was basically observed for most of the combinations of  $C_1$  and  $C_2$ . The other two cases, that is,  $n = 3$  and  $n = 7$ , show themselves as intermediary results when it comes to the success rate. Putting the results into the context, our problem comprises three maxima at most. Thus, three particles

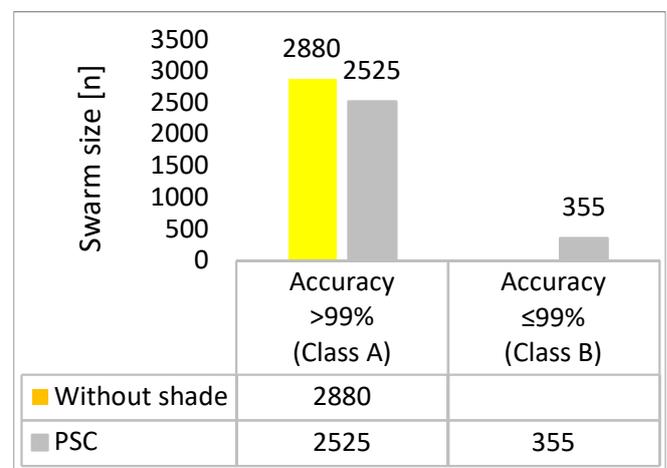
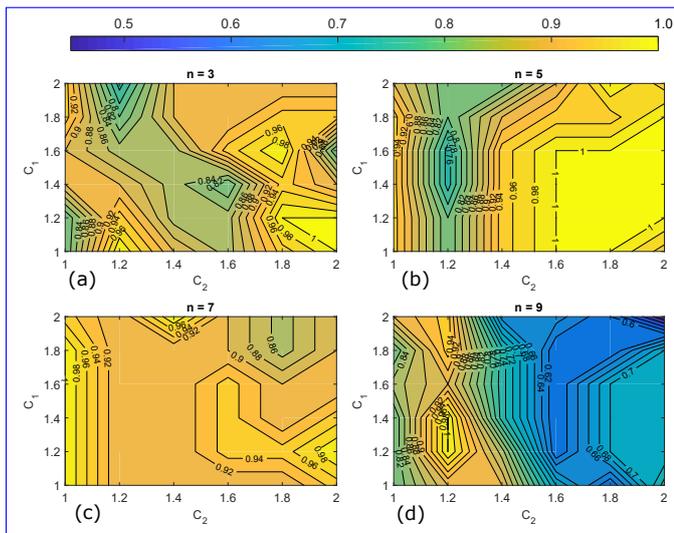


Figure 7. Accuracy and classification of simulation results independently of the settings.



**Figure 8.** Success rate mapping for different number of particles (a)  $n = 3:5$  pairs with  $SR = 100\%$ . (b)  $n = 5:13$  pairs with  $SR = 100\%$ . (c)  $n = 7:7$  pairs with  $SR = 100\%$ . (d)  $n = 9:2$  pairs with  $SR = 100\%$ .

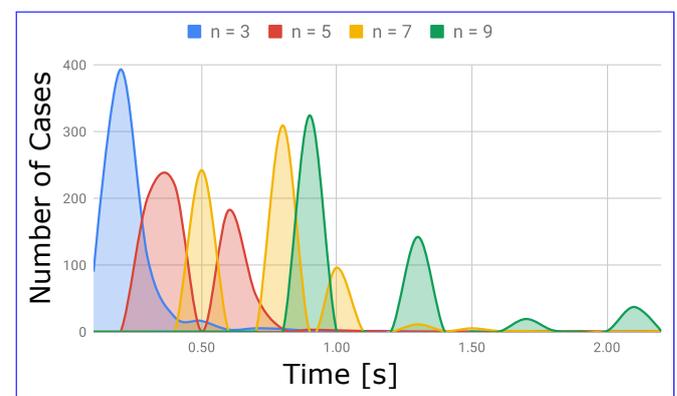
may be a small number and could lead to blind points in the searching range and reduce the success rate. On the other extremity, nine particles may exceed the limit for good results, as it could facilitate the concentration of articles around local maxima. This can be retrieved from the Figure 8 by noticing that the constant related to the social parcel,  $C_2$ , of the algorithm is inherently linked with the bad results. This is logical because the greater the number of particles, the greater is the number of particles located nearby local maxima at the beginning of the search. This makes the social component a source for disrupting the particles moving toward the GMPP. Although seven particles may seem to present more homogeneous results, with five particle numbers a success rate close to 100% was obtained for a certain range of  $C_1$  and  $C_2$ .

Concerning the convergence elapsed-time, the influence of the hyper-parameters can be analyzed through the Tables 4-7. They present the average time spent during the searching based on the results in group A only, that is, those which reach at least 99% theoretical GMPP. Before starting the analysis, it is important to notice that the range of colors are not the same for different tables due to aesthetic issues. Thus, the color which represents a certain value of elapsed time for one table may not be the same for others. Nevertheless, it was decided to use cold colors for the best results and warm ones for the worst of each number of particles.

Analyzing each table individually one may notice a convergence time strongly influenced by the number of particles. This is better illustrated in the distribution chart presented in Figure 9. It is also correct to consider

this hyper-parameter as the one that most influence the convergence time. Indeed, the lower the number of particles, the faster is the searching for the GMPP and the difference stands out. While it could be pointed out an average convergence time of roughly  $200ms$  for the case with three particles, this average rapidly grows toward  $900ms$  for the case with nine particles. Besides that, as it is possible to observe in Figure 9, the cases with greater number of particles presented more disperse distribution. In this respect, the standard deviation assume the values of 0.13, 0.18, 0.20, 0.40 for  $n = 3, 5, 7, 9$  respectively. This observation can be explained by the amount of time spent in each iteration of the swarm loop. For instance, being  $\Delta t$  the necessary time period for assessing the power produced by each particle, and considering it far way greater than any computational delay, the elapsed time for one iteration of the swarm loop for three particles would be  $3\Delta t$ . Meanwhile, for the cases of five, seven and nine particles, it is expected elapsed-times of  $5\Delta t$ ,  $7\Delta t$  and  $9\Delta t$  and this relationship can somehow noticed in the results presented in Tables 4-7. Thus, it is fair to state that, in general, the number of cycles for reaching the GMPP is nearly the same for all the different particle numbers.

The second hyper-parameter to influence the convergence dynamics is the inertia weight ( $w$ ). For any number of particles, the greater this variable is, the worse is the algorithm performance. This result is also consistent with the definition of  $w$  itself. As it represents the inertia of the particles, great values of it can slow them down when either the social or the cognitive parcels of (9) face fast changes. Notice that in any moment the statement of the previous clauses suggest that  $w$  may be chosen as zero to fasten up the process. In fact, this parameter commands the ability to move forward into the best point, and its total disabling would only give the algorithm an 100% random characteristic. As consequence, the convergence toward the MPP would be barely possible.



**Figure 9.** Convergence time of the cases classified in group A.

Table 4. Average convergence time in seconds for three particles.

		w = 0.2 (a)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.151	0.136	0.121	0.091	0.486	0.125
	1.2	0.151	0.136	0.121	0.091	0.241	0.136
	1.4	0.151	0.136	0.121	0.091	0.091	0.121
	1.6	0.251	0.136	0.121	0.106	0.189	0.159
	1.8	0.241	0.136	0.121	0.106	0.114	0.121
	2	0.302	0.146	0.121	0.106	0.114	0.121

		w = 0.4 (b)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.132	0.196	0.131	0.125	0.125	0.140
	1.2	0.132	0.256	0.131	0.200	0.125	0.163
	1.4	0.144	0.211	0.131	0.121	0.125	0.151
	1.6	0.144	0.211	0.131	0.178	0.136	0.211
	1.8	0.144	0.211	0.354	0.148	0.136	0.174
	2	0.144	0.114	0.131	0.196	0.189	0.185

		w = 0.6 (c)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.136	0.211	0.170	0.151	0.136	0.159
	1.2	0.151	0.151	0.166	0.151	0.136	0.125
	1.4	0.200	0.294	0.151	0.151	0.136	0.136
	1.6	0.189	0.121	0.148	0.136	0.148	0.148
	1.8	0.211	0.136	0.148	0.151	0.189	0.148
	2	0.260	0.136	0.159	0.151	0.159	0.148

		w = 0.8 (d)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.298	0.189	0.170	0.193	0.226	0.193
	1.2	0.161	0.181	0.170	0.223	0.283	0.181
	1.4	0.166	0.181	0.279	0.170	0.170	0.181
	1.6	0.176	0.287	0.148	0.148	0.136	0.181
	1.8	0.178	0.347	0.159	0.170	0.159	0.159
	2	0.155	0.396	0.148	0.170	0.170	0.170

		w = 1.0 (e)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.204	0.350	0.256	0.260	0.260	0.226
	1.2	0.193	0.238	0.238	0.238	0.380	0.215
	1.4	0.294	0.215	0.238	0.249	0.238	0.268
	1.6	0.193	0.215	0.241	0.365	0.260	0.372
	1.8	0.215	0.347	0.226	0.260	0.352	0.253
	2	0.249	0.332	0.200	0.302	0.241	0.251

Table 5. Average convergence time in seconds for five particles.

		w = 0.2 (a)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.345	0.251	0.251	0.251	0.251	0.251
	1.2	0.471	0.251	0.251	0.251	0.251	0.251
	1.4	0.439	0.251	0.251	0.251	0.251	0.251
	1.6	0.408	0.251	0.251	0.251	0.283	0.251
	1.8	0.439	0.251	0.251	0.251	0.283	0.251
	2	0.377	0.251	0.251	0.251	0.314	0.251

		w = 0.4 (b)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.283	0.911	0.251	0.377	0.345	0.377
	1.2	0.314	0.251	0.251	0.377	0.314	0.377
	1.4	0.314	0.314	0.251	0.377	0.314	0.345
	1.6	0.314	0.314	0.314	0.408	0.377	0.377
	1.8	0.283	0.314	0.314	1.003	0.377	0.439
	2	0.283	0.314	0.345	0.377	0.377	0.377

		w = 0.6 (c)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.314	0.377	0.418	0.377	0.439	0.439
	1.2	0.283	0.544	0.418	0.377	0.377	0.439
	1.4	0.345	0.377	0.418	0.345	0.408	0.439
	1.6	0.377	0.314	0.418	0.345	0.439	0.408
	1.8	0.377	0.377	0.418	0.377	0.439	0.408
	2	0.377	0.377	0.377	0.408	0.408	0.533

		w = 0.8 (d)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.418	0.408	0.533	0.502	0.533	0.502
	1.2	0.377	0.377	0.533	0.439	0.565	0.471
	1.4	0.418	0.439	0.565	0.439	0.502	0.471
	1.6	0.418	0.408	0.533	0.471	0.659	0.471
	1.8	0.377	0.377	0.460	0.460	0.953	0.471
	2	0.439	0.439	0.439	0.533	0.596	0.471

		w = 1.0 (e)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.502	0.502	0.502	0.502	0.502	0.533
	1.2	0.502	0.502	0.471	0.471	0.502	0.471
	1.4	0.471	0.544	0.502	0.471	0.471	0.502
	1.6	0.471	0.585	0.533	0.502	0.439	0.558
	1.8	0.502	0.565	0.565	0.502	0.533	0.502
	2	0.533	0.565	0.502	0.533	0.565	0.621

Table 6. Average convergence time in seconds for seven particles.

		w = 0.2 (a)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.676	0.492	0.492	0.574	0.615	0.615
	1.2	0.676	0.492	0.492	0.790	0.615	0.615
	1.4	0.676	0.492	0.492	0.790	0.615	0.615
	1.6	0.676	0.492	0.492	0.799	0.615	0.615
	1.8	0.676	0.492	0.492	0.799	0.615	0.615
	2	0.676	0.492	0.764	0.790	0.615	0.615

		w = 0.4 (b)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.676	0.615	0.615	0.746	0.574	0.574
	1.2	0.676	0.676	0.615	0.737	0.574	0.667
	1.4	0.676	0.676	0.615	0.851	0.574	0.656
	1.6	0.615	0.615	0.615	1.018	0.574	0.492
	1.8	0.676	0.615	0.615	0.574	0.492	0.492
	2	0.615	0.615	0.615	0.574	0.492	0.492

		w = 0.6 (c)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.492	0.553	0.615	0.574	0.574	0.615
	1.2	0.615	0.492	0.676	0.492	0.746	0.492
	1.4	0.553	0.492	0.615	0.492	0.492	0.492
	1.6	0.676	0.492	0.615	0.492	0.574	0.492
	1.8	0.676	0.492	0.615	0.492	0.492	0.553
	2	0.656	0.492	0.492	0.492	0.574	0.553

		w = 0.8 (d)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.615	0.737	0.737	0.737	0.737	0.737
	1.2	0.615	0.737	0.737	0.737	0.737	0.737
	1.4	0.737	0.737	0.737	0.737	0.676	0.737
	1.6	0.676	0.737	0.737	0.737	0.737	0.676
	1.8	0.737	0.737	0.799	0.737	0.676	0.676
	2	0.615	0.737	0.860	0.737	0.737	0.737

		w = 1.0 (e)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.799	0.983	0.922	0.983	0.983	0.922
	1.2	0.860	1.044	0.983	0.983	0.922	0.983
	1.4	0.799	0.983	0.983	0.983	0.860	0.860
	1.6	0.860	1.044	1.106	0.922	0.860	0.983
	1.8	0.922	0.922	0.983	0.860	0.860	0.860
	2	0.922	0.860	0.860	0.737	0.860	0.983

Table 7. Average convergence time in seconds for nine particles.

		w = 0.2 (a)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	1.016	0.813	1.016	0.813	0.813	0.813
	1.2	1.016	0.813	0.813	0.813	0.813	0.813
	1.4	1.016	0.813	0.813	0.813	0.813	0.813
	1.6	1.016	0.813	0.813	0.813	0.813	0.813
	1.8	1.016	0.813	0.813	0.813	0.813	0.813
	2	1.016	0.813	0.813	0.813	0.813	0.813

		w = 0.4 (b)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.813	0.813	0.813	0.813	0.813	0.813
	1.2	0.813	0.813	1.203	0.813	0.813	0.813
	1.4	0.813	0.813	0.813	0.813	0.813	0.813
	1.6	0.813	0.813	0.813	0.813	0.813	0.813
	1.8	1.117	0.813	0.813	0.813	0.813	0.813
	2	0.813	1.218	0.813	1.128	0.813	0.813

		w = 0.6 (c)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1	0.813	0.813	0.813	0.813	0.813	1.354
	1.2	0.813	0.813	0.813	0.813	1.016	0.813
	1.4	0.813	0.813	0.813	1.016	1.016	0.813
	1.6	0.813	0.813	0.813	0.813	1.016	0.813
	1.8	0.813	0.813	1.113	0.813	0.813	0.813
	2	0.813	1.117	0.813	0.813	0.813	0.813

		w = 0.8 (d)					
		C <sub>1</sub>					
		1	1.2	1.4	1.6	1.8	2
C <sub>2</sub>	1						

As for  $C_1$  and  $C_2$  they did not had an important role into the convergence dynamics. In fact, it was not noticed influence of them in such a clear way as the other two hyper-parameters. However, it was not possible to rule them out from the group of parameters that influence the dynamic performance of the MPPT. For example, consider the result for the case with  $n = 3$ ,  $w = 0.2$ ,  $C_1 = 1.0$  and  $C_2 = 1.6$ . This point is surrounded by the fastest cases obtained throughout the scenarios and still presented the slowest convergence for three particles. Special cases such as those were found in all four tables and they somehow flag a small, yet not null, influence of these constants into the convergence dynamics. Nonetheless, this influence may not be as linear as the influence of number of particles and the weight ( $w$ ). Here is fair to point out that  $C_1$  and  $C_2$  appears in the cognitive and social parcels of (9) and those are also influenced by random numbers  $r_1$  and  $r_2$ , and chances are that the atypical results discussed in the beginning of this paragraph come out of aleatory events. These could be better addressed by means of increasing significantly the number of test cases so as to reduce the weight of aleatory events into the global result.

Considering the detailed analysis presented in this section is possible to point out the case with five particles, cognitive and social components,  $C_1$  and  $C_2$ , equal to 1.6, and the inertia weight  $w$  equal to 0.2 as the best set for the algorithm. Of course the convergence time is important, by here the leading factor for this conclusion was the success rate.

It is important to notice that, in having more series-associated panels, the number of possible multiple MPPs grows and for this reason it is not possible to assure that a five-particle PSO-MPPT will lead to the best performance in these new scenarios.

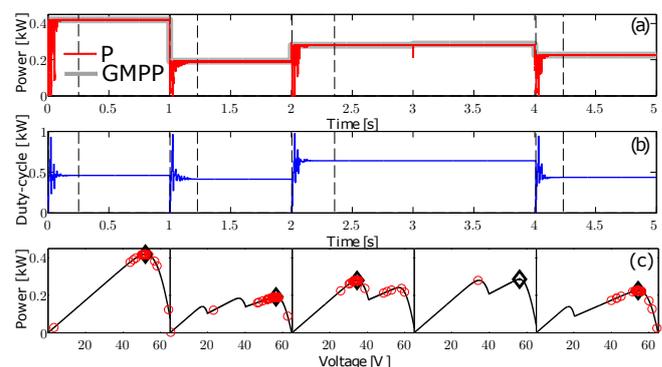
## 8. Simulation Results

The objective of this section is to observe the PSO algorithm, with the hyper-parameters adjusted according to the previous analysis, acting on the simulated circuit represented in Figure 4. For this matter, it was considered only the best case spotted in the previous section, that is, the PSO was conceived with five particles,  $w = 0.2$ ,  $C_1 = 1.6$  and  $C_2 = 1.6$ .

In Figure 10 there are the obtained results with the simulated circuit. It was considered five different patterns for the P-V curve, each one remaining for one full minute. The changing of patterns in a roll aims at observing the behaviour of the proposed MPPT both in transient- and steady-state conditions. Figure 10(a), for instance presents the produced power and the theoretical GMPP respectively in red and gray lines, whereas Figure 10(b) shows the corresponding duty-cycle of the dc-dc converter. For making it more

comprehensive, the used patterns for the P-V curve, along with particles loci, are presented in Figures 10(c), precisely displayed so as to indicate at which time interval they correspond to. In each of these graphs, the diamond-shaped parks indicate the GMPP, whereas the circles indicate the particles loci. Besides that, the temperature was held constant in 25°C and Table 8 presents the average irradiation in each panel that produces the aforementioned P-V patters.

In the first test case, it was considered the panels without PSC and facing the maximum irradiation considered in the simulations. Thus, the power produced in the first minute was barely close to 420W, which corresponds to the highest produced power of the entire simulation. The graph in the Figure 10(a) shows that the algorithm presented no difficult to reach the single MPP or to reach the newer GMPP when the the panels started presenting PSC in  $t = 1min$ . At  $t = 2min$  the shading pattern changed again and once more the PSO-MPPT reached the new GMPP. Here it is important to notice the significant difference between this new maximum and the previous one. The algorithm was only able to reach the new maximum because after leaving the halt loop, the particles are spread over the entire domain before starting the searching. At  $t = 3min$  it was carefully chosen a pattern which would not cause a changing in the produced power enough for starting a new swarm loop in the algorithm. Thus, the PSO-MPP remained in the halt loop and the system did not find the new GMPP, what can be confirmed by seeing the particles loci in the corresponding segment of Figure 10(c). In this case, however, there was no significant loss in the generation because the difference between the local and the global maxima is roughly 0.14%, what can be observed in Table 8. It is importance



**Figure 10.** Simulation results for different transient events considering  $n = 5$ ,  $w = 0.2$ ,  $C_1 = 1.6$  and  $C_2 = 1.6$ . (a) Produced power (b) Duty cycle. (c) P-V pattern adopted in each time interval. The colors blue, red were used respectively for the instantaneous duty-cycle and power and the gray for the theoretical GMPP. In (c) the diamond-shaped marks indicate the GMPP and the red circles the particle loci.

**Table 8.** Evaluation of the effectiveness of the MPPT for each P-V curve in the simulation

Time Interval	G1	G2	G3	GMPP	$P_{pso}$	Efficiency	Convergence time
From 0 to 1min	1000W/m <sup>2</sup>	1000W/m <sup>2</sup>	1000W/m <sup>2</sup>	419.97W	419.45W	99.88%	251.5ms
From 1 to 2min	400W/m <sup>2</sup>	600W/m <sup>2</sup>	1000W/m <sup>2</sup>	191.73W	190.61W	99.42%	228.4ms
From 2 to 3min	1000W/m <sup>2</sup>	1000W/m <sup>2</sup>	500W/m <sup>2</sup>	279.98W	279.49W	99.82%	355.7ms
From 3 to 4min	1000W/m <sup>2</sup>	600W/m <sup>2</sup>	1000W/m <sup>2</sup>	285.77W	279.49W	97.80%	-
From 4 to 5min	1000W/m <sup>2</sup>	500W/m <sup>2</sup>	500W/m <sup>2</sup>	225.32W	224.75W	99.75%	234.4ms

to notice that the percent of change in the measured power considered for sparking the swarm loop (5% in this work) plays a key role at the higher divergence from the GMPP the system may face. Of course it is possible to narrow this band reducing the mentioned percentage, yet it can make the system prone to chatter due noises in the measurements (in this case the noise could be misinterpreted as a variation in the power and the searching for the GMPP restarted). Thus, here we have a tread off between noise immunity and capacity to converge to the GMPP in all the cases. Finally, at  $t = 4min$  the last P-V pattern started and, as the MPPT once more converged to the GMPP. All in all, the MPPT allowed the system to reach the GMPP in all simulated P-V patterns in no more than 360ms and this is registered in Table 8.

Still on Table 8, it is worthwhile noticing that the steady-state power matches the theoretical GMPP with nearly 100% precision, reinforcing the effectiveness of the used PSO-MPPT with the introduced way to adjust the hyper-parameters.

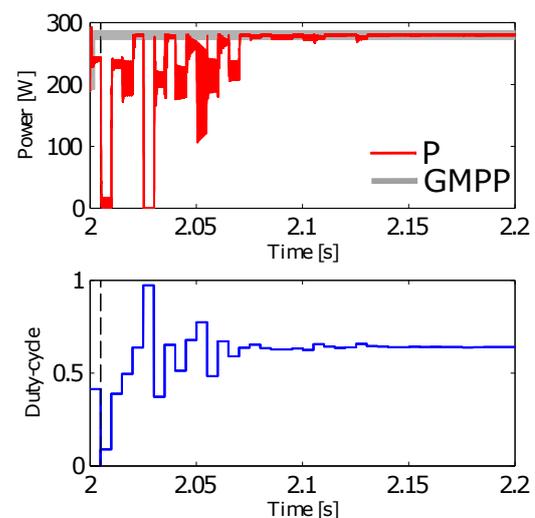
An other point to be noted in the Figure 10 is the chattered pattern observed in the duty cycle and, consequently, in the produced power just after each changing of the P-V characteristic. This can be better observed in Figure 11 and is basically caused by assessment of the power produced by each particle. In other words, when the system faces a change in the produced power great enough for the enabling the swarm loop in the algorithm, the MPPT drives the converter with the duty cycle of each particle, season over season of the swarm loop, causing this chattered characteristic in  $d$  and  $P$ . Once the particles converge to the MPP, the halt loop takes place and both the duty cycle and the produced power remain constants at  $d_{best,g}$  and  $P_{max,g}$ , respectively.

## 9. Conclusions

This paper presented a general review of the use of PSO in MPPT with a methodology to adjust the hyper-parameters such that the implemented algorithm was capable of identifying, correctly, the GMPP in more than 99% of the cases. In a nutshell, it has been presented a grid search over a space of 5760 scenarios for the best combination of parameters, number of particles,  $C_1$ ,  $C_2$  and  $w$ , for the proposed algorithm.

From the results, it was possible to conclude that the algorithm is effective and reached the GMPP in almost 88% of proposed cases independently of the settings (number of particles and chosen values for  $C_1$ ,  $C_2$  and  $w$ ). It was also noticed that the rate of non-convergence, which in fact, for this analysis, means not reaching the GMPP before  $t = 3s$ , was low with only 39 out of 5760 cases reporting this behavior. Of course, the algorithm was able to reach the GMPP nearly 100% of the cases for  $n = 5$ ,  $C_1 = C_2 = 1.6$  (best setting arrangement). It was also spotted that the best performances, when it comes to the convergence time, led the algorithm to reach GMPP in about 0.25s. Besides that, it is also possible to draw some conclusions about the relationship between parameter adjustment and tracking performance:

- Swarm size ( $n$ ): large number of particles leads to a slower convergence time.
- Inertia coefficient ( $w$ ): larger value can slow the convergence time since it might require more cycles for all particles get closer to the GMPP.
- Cognitive and social components: inside the algorithm  $C_1$  and  $C_2$  contributes either for local and global maximum searching.



**Figure 11.** Characteristic chattering observed in PSO. The colors blue, red were used respectively for the instantaneous duty-cycle and power and the gray for the theoretical GMPP.

One last point to be made concerns the effect of the general configuration on the results achieved. Since in the present analysis was considered only cases with one, two or three local maxima, it is not possible to guarantee that the performance will be the same for the cases where more than three maxima occurs. In this case a novel database of different simulation test cases must be considered to perform the hyper-parameters adjustment.

## References

- [1] MALINOWSKI, M., LEON, J.I. and ABU-RUB, H. (2017) Solar photovoltaic and thermal energy systems: Current technology and future trends. *Proceedings of the IEEE* **105**(11): 2132–2146. doi:[10.1109/jproc.2017.2690343](https://doi.org/10.1109/jproc.2017.2690343).
- [2] ARANTEGUIL, R.L. and JÄGER-WALDAU, A. (2018) Photovoltaics and wind status in the european union after the paris agreement. *Renewable and Sustainable Energy Reviews* **81**: 2460–2471. doi:[10.1016/j.rser.2017.06.052](https://doi.org/10.1016/j.rser.2017.06.052).
- [3] YU, H.J.J., POPIOLEK, N. and GEOFFRON, P. (2014) Solar photovoltaic energy policy and globalization: a multiperspective approach with case studies of germany, japan, and china. *Progress in Photovoltaics: Research and Applications* **24**(4): 458–476. doi:[10.1002/pip.2560](https://doi.org/10.1002/pip.2560).
- [4] BENDIB, B., BELMILI, H. and KRIM, F. (2015) A survey of the most used mppt methods: Conventional and advanced algorithms applied for photovoltaic systems. *Renewable and Sustainable Energy Reviews* **45**: 637–648. doi:[10.1016/j.rser.2015.02.009](https://doi.org/10.1016/j.rser.2015.02.009).
- [5] LIU, F., KANG, Y., ZHANG, Y. and DUAN, S. (2008) Comparison of p& o and hill climbing mppt methods for grid-connected pv converter. In *2008 3rd IEEE Conference on Industrial Electronics and Applications*: 804–807. doi:[10.1109/ICIEA.2008.4582626](https://doi.org/10.1109/ICIEA.2008.4582626).
- [6] RODRIGUEZ, E.A., FREITAS, C.M., BELLAR, M.D. and MONTEIRO, L.F.C. (2015) Mppt algorithm for pv array connected to a hybrid generation system. In *2015 IEEE 24th International Symposium on Industrial Electronics (ISIE)*: 1115–1120. doi:[10.1109/isie.2015.7281628](https://doi.org/10.1109/isie.2015.7281628).
- [7] MONTEIRO, L.F.C., FREITAS, C.M. and BELLAR, M.D. (2019) Improvements on the incremental conductance mppt method applied to a pv string with single-phase to three-phase converter for rural grid applications. *Advances in Electrical and Computer Engineering* **19**: 63–70. doi:[10.4316/AECE.2019.01009](https://doi.org/10.4316/AECE.2019.01009).
- [8] LIU, L., MENG, X. and LIU, C. (2016) A review of maximum power point tracking methods of pv power system at uniform and partial shading. *Renewable and Sustainable Energy Reviews* **53**: 1500–1507. doi:[10.1016/j.rser.2015.09.065](https://doi.org/10.1016/j.rser.2015.09.065).
- [9] KHARE, A. and RANGNEKAR, S. (2013) A review of particle swarm optimization and its applications in solar photovoltaic system. *Applied Soft Computing* **13**(5): 2997 – 3006. doi:[10.1016/j.asoc.2012.11.033](https://doi.org/10.1016/j.asoc.2012.11.033).
- [10] ISHAQUE, K., SALAM, Z., AMJAD, M. and MEKHILEF, S. (2012) An improved particle swarm optimization (psa)-based mppt for pv with reduced steady-state oscillation. *IEEE transactions on Power Electronics* **27**(8): 3627–3638. doi:[10.1109/tpel.2012.2185713](https://doi.org/10.1109/tpel.2012.2185713).
- [11] RENAUDINEAU, H., DONATANTONIO, F., FONTCHASTAGNER, J., PETRONE, G., SPAGNUOLO, G., MARTIN, J.P. and PIERFEDERICI, S. (2015) A pso-based global mppt technique for distributed pv power generation. *IEEE Transactions on Industrial Electronics* **62**(2): 1047–1058. doi:[10.1109/tie.2014.2336600](https://doi.org/10.1109/tie.2014.2336600).
- [12] DE OLIVEIRA, F.M., DA SILVA, S.A.O., DURAND, F.R., SAMPAIO, L.P., BACON, V.D. and CAMPANHOL, L.B. (2016) Grid-tied photovoltaic system based on pso mppt technique with active power line conditioning. *IET Power Electronics* **9**(6): 1180–1191. doi:[10.1049/iet-pel.2015.0655](https://doi.org/10.1049/iet-pel.2015.0655).
- [13] KOAD, R.B., ZOBAA, A.F. and EL-SHAHAT, A. (2017) A novel mppt algorithm based on particle swarm optimization for photovoltaic systems. *IEEE Transactions on Sustainable Energy* **8**(2): 468–476. doi:[10.1109/tste.2016.2606421](https://doi.org/10.1109/tste.2016.2606421).
- [14] SEN, T., PRAGALLAPATI, N. and AND, V.A. (2018) Global maximum power point tracking of pv arrays under partial shading conditions using a modified particle velocity-based pso technique. *IET Renewable Power Generation* **12**: 555–564. doi:[10.1049/iet-rpg.2016.0838](https://doi.org/10.1049/iet-rpg.2016.0838).
- [15] MARQUES LEOPOLDINO, A.L., MAGALHÃES FREITAS, C. and CORRÊA MONTEIRO, L.F. (2019) On the effects of parameter adjustment on the performance of pso-based mppt of a pv-energy generation system. In AFONSO, J.L., MONTEIRO, V. and PINTO, J.G. [eds.] *Green Energy and Networking* (Cham: Springer International Publishing): 175–192. doi:[10.1007/978-3-030-12950-7\\_14](https://doi.org/10.1007/978-3-030-12950-7_14).
- [16] VILLALVA, M.G., GAZOLI, J.R. and FILHO, E.R. (2009) Comprehensive approach to modeling and simulation of photovoltaic arrays. *IEEE Transactions on Power Electronics* **24**(5): 1198–1208. doi:[10.1109/tpel.2009.2013862](https://doi.org/10.1109/tpel.2009.2013862).
- [17] DOLARA, A., LEVA, S. and MANZOLINI, G. (2015) Comparison of different physical models for pv power output prediction. *Solar Energy* **119**: 83–99. doi:[10.1016/j.solener.2015.06.017](https://doi.org/10.1016/j.solener.2015.06.017).
- [18] HASAN, M. and PARIDA, S. (2016) An overview of solar photovoltaic panel modeling based on analytical and experimental viewpoint. *Renewable and Sustainable Energy Reviews* **60**: 75–83. doi:[10.1016/j.rser.2016.01.087](https://doi.org/10.1016/j.rser.2016.01.087).
- [19] SHEPARD, N. (1984) *Diodes in Photovoltaic Modules and Arrays*. Final report, Prepared for JPL by General Electric Company Advanced Energy Systems and Technology Division, King of Prussia, Pennsylvania. March 15, 1984.
- [20] KENNEDY, J. and EBERHART, R. (1995) Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, **4**: 1942–1948. doi:[10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [21] SONG, M.P. and GU, G.C. (2004) Research on particle swarm optimization: a review. In *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, **4**: 2236–2241. doi:[10.1109/ICMLC.2004.1382171](https://doi.org/10.1109/ICMLC.2004.1382171).
- [22] MIRHASSANI, S.M., GOLROODBARI, S.Z.M., GOLROODBARI, S.M.M. and MEKHILEF, S. (2015) An improved particle swarm optimization based maximum power point tracking strategy with variable sampling time. *International Journal of Electrical Power & Energy Systems* **64**: 761–770. doi:[10.1016/j.ijepes.2014.07.074](https://doi.org/10.1016/j.ijepes.2014.07.074).