energies

MDPI

Article

Maximum Power Point Tracking Based on Reinforcement Learning Using Evolutionary Optimization Algorithms

Kostas Bavarinos [1], Anastasios Dounis [2,*] and Panagiotis Kofinas [1,2]

[1] Industrial Design and Production Engineering, University of West Attica, 250 Thivon & P. Ralli Str, 12241 Egaleo, Greece; Bavarinos@gmail.com (K.B.); pkofinas@uniwa.gr (P.K.)
[2] Biomedical Engineering, University of West Attica, Ag. Spyridonos 17, 12243 Egaleo, Greece
* Correspondence: aidounis@uniwa.gr

Abstract: In this paper, two universal reinforcement learning methods are considered to solve the problem of maximum power point tracking for photovoltaics. Both methods exhibit fast achievement of the MPP under varying environmental conditions and are applicable in different PV systems. The only required knowledge of the PV system are the open-circuit voltage, the short-circuit current and the maximum power, all under STC, which are always provided by the manufacturer. Both methods are compared to a Fuzzy Logic Controller and the universality of the proposed methods is highlighted. After the implementation and the validation of proper performance of both methods, two evolutionary optimization algorithms (Big Bang—Big Crunch and Genetic Algorithm) are applied. The results demonstrate that both methods achieve higher energy production and in both methods the time for tracking the MPP is reduced, after the application of both evolutionary algorithms.

Keywords: maximum power point tracking; reinforcement learning; q-learning; state–action-reward-state–action; evolutionary algorithms; optimization; fuzzy logic controller

Citation: Bavarinos, K.; Dounis, A.; Kofinas, P. Maximum Power Point Tracking Based on Reinforcement Learning Using Evolutionary Optimization Algorithms. *Energies* **2021**, *14*, 335. https://doi.org/10.3390/en14020335

Received: 27 October 2020
Accepted: 7 January 2021
Published: 9 January 2021

1. Introduction

The constantly rising demand for electricity and the necessity to attend the carbon emission problem has given growth to the exploitation of renewable energy sources (RES). Although RES cannot yet meet the world's power demands, a contribution to the existing conventional energy sources (fossil fuel, natural gas, nuclear) presents a beneficial option, both environmentally and economically.

In recent years, many studies have been focused on solar energy and more specifically, photovoltaic (PV) systems, which convert the sun's electromagnetic radiation into electricity. Despite the fact that PVs produce clean energy and the sun is considered an inexhaustible energy source, they come with two major drawbacks. The first one is the poor conversion efficiency of sun's insolation, while the second one is the fact that the electrical load connected to the system defines its operating power point. Additionally, the nonlinear curves of power–voltage (P–V) and current–voltage (I–V) due to the varying atmospheric conditions, such as temperature and irradiance, add additional complexity. Since there is a maximum power point (MPP) for any given pair of irradiance-temperature, the design and implementation of a Maximum Power Point Tracking (MPPT) controller, that forces the PV source to operate at the MPP at any time for the given environmental conditions, is crucial in order to maximize the efficiency of PVs.

A wide variety of MPPT techniques can be found in the literature [1]. These techniques are classified as direct or indirect methods. The former ones require real-time measurements from the PV array to be extracted, such as voltage, current or power, while the latter ones rely on the parameters of the PV source, along with data of its operating conditions, in order to create a relatively accurate mathematical model for the PV array. The parameters of the PV source usually are the open—circuit voltage (V_{OC}) and the short—circuit current I_{SC}. Popular indirect methods include Open—Circuit Voltage (OCV) [1] and Short—Circuit

Current (SCC) [1]. These methods, while simple enough, can never achieve true MPPT since they are based on approximations between V_{OC}, I_{SC} and V_{MPP}, I_{MPP}, respectively. Furthermore, the measurement of V_{OC} and I_{SC} cause a periodically complete power loss to occur. On the other hand, direct methods, although can be complex, exhibit superior performance in most cases. Some of them are Perturb and Observe (P&O) methods [2–6], incremental conductance (IC) methods [7,8], neural networks (NN) methods [9–12], and fuzzy logic (FL) methods [13–17]. P&O method seems to be one of the most frequently used techniques due to its reliability, simplicity and easiness to implement. However, the P&O is based on the perturbation of PV's array voltage, engenders oscillations around the MPP. In addition, fluctuating environmental conditions might result in power losses. Despite its shortcomings, P&O modifications can tackle these problems [4,5]. IC method offers smoothness when it comes to tracking the MPP under changing environmental conditions. Nevertheless, the control process is somewhat complex and the quality of the measurements affects its performance in some measure. NN methods offer fast MPP tracking speed and high efficiency. Having said that, the training of the NN can be time consuming and the accuracy of the results is affected by the number of neurons present at the hidden layer. FL methods indicate high convergence speed to MPP without oscillations around it. Complex implementation of these methods constitutes one of their problem. The other one is the fact that the control engineer's choice of fuzzy sets, membership functions' shape and development of rule tables affects their performance greatly.

Many metaheuristic optimization algorithms, for instance particle swarm optimizer (PSO) and genetic algorithms (GA) have been employed to improve the performance of various MPPT techniques [18,19]. In Ref. [16] an optimization algorithm called BB—BC is applied to improve the parameters of a Fuzzy PID controller and the aforementioned disadvantages of FL methods. In [20] the algorithm optimizes the membership functions of a fuzzy controller. According to the author in [21], BB—BC is capable of finding the optimal solution with high speed and convergence and therefore, is superior to other optimization techniques.

A few researchers have proposed recently Reinforcement Learning based methods to address the MPPT problem and deal with the limitations that current MPPT techniques present [22–25] such as oscillations around the MPP that P&O produces, complexity in the control process of IC, time required for the training of NN methods and the affection of the designer's choices in FL methods. In Ref. [23] an RL MPPT controller is designed that uses a set of seven actions and four states, each one representing the movement direction of the operating point compared to that of the MPP. A similar approach is being taken in Ref. [25] with a set of four actions and the same states as described in the previous reference. Both of these methods achieve MPPT but small oscillations around the MPP are present. Also, the first one requires some significant time to finish the exploration phase, while the latter one is tested only under varying irradiance conditions. In Ref. [24] two RL approaches are considered. The RL-QT MPPT control method exhibits good performance and no oscillations at the MPP, but the size of the Q table consists of 29,820 state–action pairs and additional hardware is required for the measurement of irradiance and temperature, other than voltage and current.

In previous work [22] a Q-Learning model was proposed which can track the MPP under different environmental conditions and PV sources. This paper seeks to address the aforementioned problems using two RL approaches, and more specifically, the Q-learning and SARSA algorithms by using five possible actions and a Q table of 4000 state–action pairs to make the method computationally efficient. The proposed methods exhibit no oscillations once MPP is reached. Furthermore, the RL MPPT methods are applied in different PV sources to examine their performance and demonstrate their functionality under a diverse set of PVs. These RL approaches require several parameters to be defined, for example, the amount of exploration rounds, the learning rate α and discount factor γ and the actions. In most cases these are set up by trial and error method. This paper aims to optimize all those parameters using BB—BC algorithm for a specific PV source, to

further improve the power extracted at every moment, and afterwards use them on PVs with different electrical characteristics to inspect their effect. This paper contributes to the existing literature as follows:

- Two universal Reinforcement Learning algorithms for the MPPT problem are being applied and the only necessary knowledge of the PV's characteristics are V_{OC}, I_{SC}, and P_{MPP} under STC, which are given by the manufacturer's datasheet;
- Results and graphs are being provided, verifying that both the algorithms succeed in MPPT under varying irradiance and temperature conditions, for different PV systems;
- The RL methods are compared to a Fuzzy Logic Controller for different sources in order to be highlighted the universality of the proposed methods;
- Employment of offline BB—BC optimization algorithm showcases that the extracted power can be improved even further;
- The optimized parameters of the two RL algorithms are being applied in different PV sources with even better results and are also compared with results occurring from the application of a genetic algorithm; and lastly
- Results of Q-learning and SARSA algorithms are compared to determine their MPPT control performance.

To sum up, this paper is divided into 4 sections. Section 2 presents the basic theory of PV source operation and reinforcement learning, the algorithms used and the configuration of the simulations. The results can be found in Sections 3 and 4 provides focuses on discussing the results but also, some future research directions that the authors would like to focus on.

2. Materials and Methods

2.1. PV Source Operation

As mentioned before, PVs convert sun's electromagnetic radiation into electricity. Sunlight contains photons that can excite the electrons of a PV source, causing them to move to the conduction band and resulting in the generation of current. However, the Sun's energy is not efficiently converted to electricity for a variety of reasons. William Shockley and Hans Queisser defined the maximum conversion efficiency of a single p-n junction solar cell at 33.7% [26].

Furthermore, a PV is characterized by the P-V and I-V nonlinear curves. A typical I-V curve is presented in Figure 1. The resistive value of the connected electrical load defines the operating point of the PV source. As the ohmic value of the electrical load decreases, the operating points shifts towards the left of the MPP and vice versa. When the value of the load equals V_{MPP}/I_{MPP} then the operating point coincides with the MPP and maximum power extraction is achieved. In case that a different load is used, the PV will not deliver the maximum possible power, so a MPPT controller should be applied.

Connecting the ideal load to a PV source is impractical and, in many cases, infeasible, due to the possible dynamic nature of the load. Moreover, the I–V curve changes over time because of the degradation of solar cells and the operating point of an ideal load will slowly move away from the MPP.

There are also two parameters that affect the I–V curve greatly. The first one is irradiance. I_{SC} is proportional to it and V_{oc} decreases slightly as irradiance decreases. The second parameter is the temperature of the PV source. Higher temperature values increment slightly I_{SC} while they reduce V_{OC} considerably. The increment of temperature means that electrons of the semiconductor material used in PV sources, require less energy to be excited, since a portion is acquired through thermal energy. This way, more electric current carriers (electrons) move to the conduction band and therefore, I_{SC} increases.

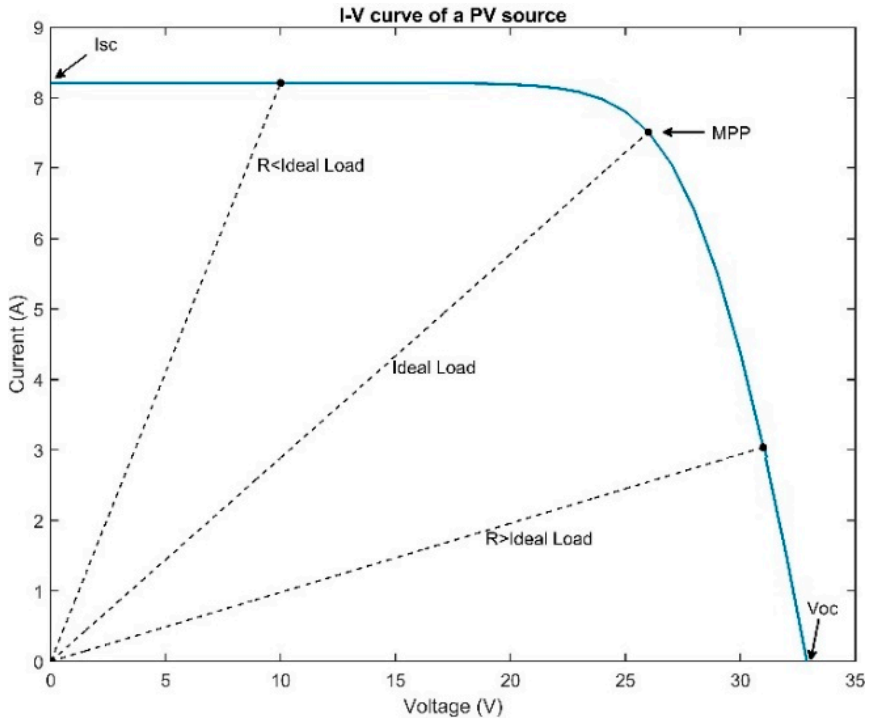


Figure 1. Typical current–voltage (I–V) curve of a photovoltaic (PV) source for constant environmental conditions.

The above-stated issues necessitate the design of MPPT control methods that force the PV source to operate at the MPP for any set of environmental conditions. Fast response time, high accuracy, quick convergence and steady performance at the MPP, low cost, and complexity, are key characteristics of these methods. Most of them include a dc–dc converter that connects between the load and the PV source. In this paper a buck (or step-down) converter is considered. The output voltage of a buck is given by Equation (1):

$$V_{out} = DV_{in} \tag{1}$$

where V_{out} is the output voltage, V_{in} is the input voltage and D is the duty cycle of the converter, ranging between 0 and 1. In an ideal scenario, where the buck converter operates with 100% efficiency, the output power of the converter must equal its input power, at any moment:

$$P_{out} = P_{in}. \tag{2}$$

Equation (2) can also be written as:

$$\frac{V_{out}^2}{R_{Load}} = \frac{V_{in}^2}{R_{epv}} \tag{3}$$

where R_{epv} is the effective resistance of the PV source [27], considering that in our case V_{in} is the PV voltage and using Equation (1), we get:

$$\frac{D^2 V_{PV}^2}{R_{Load}} = \frac{V_{PV}^2}{R_{epv}} \tag{4}$$

which leads to Equation (5):

$$R_{epv} = \frac{R_{Load}}{D^2}. \tag{5}$$

Equation (5) gives the effective resistance $R_{epv}(D, R_{Load})$ of the PV source [27] which determining the operating point of the PV source. As the duty cycle decreases, the value of the overall resistance is increased and the operation point is moving right towards the I–V characteristic curve. By adjusting the duty cycle, the operating point can be relocated to the MPP. However, a buck converter cannot be used for loads that their resistive values exceeds the resistive value of an ideal load (Figure 1). In this case the operation point will be right to the MPP and the buck converter has only the ability to move the operation point to the right direction. For this kind of loads a different type of converter must be applied like boost or buck boost converter.

Additionally, the PV model that is used in this paper is the one diode and the equations used for the generated current of the source are given in (6) to (10):

$$I_{PV} = I_{SC} - a_1 e^{b_1 V_{PV}} \tag{6}$$

$$I_{SC} = I_{scr} \frac{G_{PV}}{G_r} [1 + n_{iscT}(T_{PV} - T_r)] \tag{7}$$

$$a_1 = I_{scr} e^{-b_{STC} V_{OC}} \tag{8}$$

$$b_1 = \frac{b_{STC}}{1 + n_{vocT}(T_{PV} - T_r)} \tag{9}$$

$$b_{STC} = \frac{\log\left(1 - \frac{I_{mppr}}{I_{scr}}\right)}{V_{mppr} - V_{ocr}} \tag{10}$$

where I_{PV} and V_{PV} are the current and voltage generated from the PV source, respectively, G_{PV} is the solar irradiance received by the PV source, G_r is the reference solar irradiance at STC and equals to 1000 W/m^2, T_r is the reference temperature at STC and equals to 25 °C, T_{PV} is the temperature of the PV source, n_{vocT} is the temperature coefficient of the open circuit voltage, n_{iscT} is the temperature coefficient of the short circuit current, I_{mppr} is the current of the PV source at MPP and I_{scr} is the short circuit current, both under STC, and I_{SC} is the short circuit current for the given environmental conditions.

2.2. Reinforcement Learning

Reinforcement learning [28,29] constitutes a machine learning sector and is considered a heuristic method. Wide range of applications where data are not available or are difficult to come by utilize RL. RL problems involve an agent and its environment. The agent performs an action a_t from a given state s_t, goes to the next state s_{t+1} and receives a numeric reward R from the environment. The reward signal provided by the environment has higher value for good actions than for bad ones. The agent's goal is to maximize the accumulated reward over a long period of time. In order an agent develops a satisfactory policy, that is an action selection strategy for every given state, it has to perform some random actions from every state and review the reward signal. Nonetheless, more reward is being accumulated when the knowledge that has been gained from previous random actions is exploited. So, a balance between exploration and exploitation has to exist, otherwise the agent will fail at its task. This constant interaction between the agent and the environment allows the agent to learn and achieve its goal.

The RL problem for MPPT control can be modeled as a Markov Decision Process (MDP). States are considered to possess the Markov property when the outcome from taking an action is only affected by the current state and not any prior states, actions or rewards. If all the states of the environment are Markov states, then the environment is also a Markov environment. A RL problem is a MDP when it operates in a Markov environment.

2.2.1. States

States in RLMPPT problem inform the agent about the operating point of the PV source at any given time step. Therefore, it is important that the state space is large enough

to provide all the necessary knowledge to the agent and allow him to distinguish between different operating conditions. Nonetheless, the state space should be kept as small as possible to make the method computationally efficient. Lastly, the set of states for a RL model must be discrete.

2.2.2. Actions

A discrete set of actions exists in every MDP solving algorithm. In the case of MPPT control, the actions performed by an agent must somehow shift the operating point of the PV source. As explained previously, the operating point can be shifted by increasing or decreasing the duty cycle of the power converter, hence the action set that an agent chooses from must include positive and negative values. Negative action values reduce the duty cycle of the converter while the opposite happens for positive action values. It is also crucial that actions that cause small fluctuations of the duty cycle are included in order the operating point to be as close as possible to the MPP. With all the above in mind, the action list can be defined as $A = \{\Delta D_1, \Delta D_2, \cdots, \Delta D_n\}$, where ΔD is the change of the duty cycle and n is the number of the available actions and can vary depending on the desired performance of the MPPT controller. Larger action sets might potentially offer faster and more accurate MPPT performance at the cost of more state–action pairs of the Q-table.

2.2.3. Reward

Reward signal sent to the agent from the environment plays a vital role in an agent's learning phase. In most cases, reward is defined in such a way that is positive when good actions are chosen and negative or zero, otherwise. A negative reward signal for bad actions could encourage the agent to seek the MPP faster. Thusly, the agent will be able to distinguish between positive and negative actions. In [23] the reward for achieving an operating point very close to the MPP is set to 10, and 0 in any other case. In [22] the reward signal is defined as ΔP. This way, a positive reward value is provided to the agent for actions that increase power output of the PV source, while actions that reduce the power output provide a negative reward value to the agent.

In this paper, the proposed reward signal when the PV source operates at MPP is 1. The reward values when that is not the case are normalized and range between 0 and 1 for positive ΔP values, and between -1 and 0 for negative ΔP values.

2.2.4. Balancing Exploration and Exploitation

The agent's policy determines which action is performed in every state. Finding an optimal policy will allow the agent to maximize the accumulated long-term reward. The optimal policy can be extracted by allowing the agent enough exploration rounds, in which random actions are performed and thereafter are evaluated using the reward signal. Yet, too many exploration rounds mean that the agent is taking actions that are not necessarily the best for the state that he's located and does not exploit the knowledge that has already gathered from previous exploration rounds. So, an action-selection strategy is needed to ensure that balance between exploration and exploitation exists.

The ε-greedy selection strategy represents one simple approach to solve the problem of balance. When ε-greedy is employed, the probability of performing the action with the best outcome is $1 - \varepsilon$, while the probability to choose a random exploration action is ε, where $0 \leq \varepsilon \leq 1$. Although simple, this strategy comes with some setbacks. One of them, that will affect the performance of MPPT method, is the fact that after a good or optimal policy has been found, the agent will perform a random action ε of the time, which will entail a power loss. In addition, it is desirable that the agent will perform more exploration when a new state is presented and less when the agent comes across a state that has been revisited several times. To overcome the aforementioned issues, a modified version of the ε-greedy selection strategy is proposed.

2.3. Q-Learning and SARSA Algorithms

This subsection aims to briefly explain some basic aspects of one-step Q-learning and one-step SARSA algorithms. These algorithms are very similar and the major difference between them is the policy update rule. Q-learning uses an off-policy update rule, meaning that it does not follow the policy that it updates and the cost of exploration is not taken into account. The one step Q-learning uses the following update rule [26]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a\left[R + \gamma \max_{\alpha_{t+1}}(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \tag{11}$$

where $0 \leq a \leq 1$ is the learning rate, $0 \leq \gamma \leq 1$ is the discount factor and R is the reward. A higher discount factor value will allow the method to focus on the long-term reward rather that the short-term. Also, the learning rate should be kept relatively low in order to allow steady convergence of the learning. The usage of Equation (11) forms a so-called Q-table with state–action pairs and once its complete, the optimal policy is exported. It's worth noting that the Q-table is updated every time the agent performs an action and even after the optimal policy has been found.

On the other hand, an on-policy update rule is being employed in SARSA algorithm. Thus, the cost of exploration is taken into consideration. The update rule for the one-step SARSA algorithm is given by (12):

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a[R + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \tag{12}$$

The procedure of forming the Q-table is the same for both algorithms. The only difference between them is the update rule that is being followed in every step. The different update rule enables SARSA to learn faster than the Q-learning algorithm. However, this makes SARSA a more conservative algorithm and the probability of finding the optimal policy is higher for Q-learning. Details about the implementation of the algorithms will be given in Section 2.5.

2.4. Evolutionary Algorithms

Optimization algorithms aim to find the optimal solution of a problem by trying different parameter values and continuously simulate the problem.

2.4.1. Big Bang—Big Crunch Algorithm

Big Bang—Big Crunch (BB—BC) is comprised of two stages, namely the big bang, in which random points are generated inside a search space, and the big crunch which forces those points to converge to an optimal one [21]. The steps that compose BB—BC algorithm are:

- Step 1: Creation of initial population N representing possible solutions of the optimization problem.
- Step 2: Evaluation of each candidate solution using a fitness function. The best fitness individual is chosen as the center of mass $\vec{x}^{c}$.
- Step 3: Creation of new candidate solutions around the center of mass, using Equation (13):

$$\vec{x}^{new} = \vec{x}^{c} + \frac{lr}{k} \tag{13}$$

  where l is the parameter upper limit, r is a normal number and k is the iteration step.
- Step 4: Return to step 2 until stopping criteria have been met.

In every iteration, new candidate solutions are spread around the best fitness individual of the previous iteration and at the same time, around the entire search space. In this manner, the best solution is being constantly improved while the entire search space is being scanned for a better one. As the iteration step increases, BB—BC will focus more on searching near the best, so far, solution. Furthermore, the radius of the search space will get smaller, fact that will allow the algorithm to converge to the optimal point.

2.4.2. Genetic Algorithm

GA is an evolutionary algorithm inspired by the process of natural selection based on the abstraction of Darwin's evolution of biological systems. Genetic algorithm uses genetic operators such as crossover and recombination, mutation, and selection. The main concept of this algorithm is that a group of coded candidate solution called population is evolving toward better solutions. The coded candidate solution is called chromosome. Each individual solution is evaluated via an objective function. The better individuals are stochastically selected to combine new candidate solutions through crossover and a mutation with determinate probability is applied over the chromosomes. Through this process new population is formed and the aforementioned process is repeated for a finite number of iterations. In this paper, the GA toolbox of MATLAB is used for optimize the parameters of the proposed algorithms.

2.5. Simulation Configuration

This chapter focuses on providing the exact simulation configuration, including the parameters selected for the PV sources, the DC-DC converter and the Q-learning, SARSA and BB—BC algorithms. All the simulations were run using MATLAB and Simulink R2015a.

2.5.1. PV Sources, DC-DC Converter, Load, and PWM Signal

In many cases, PVs are connected in series or in parallel to increase the output voltage and current, respectively. So, one of the targets of this paper was the design of an MPPT control method that could be applied in PV systems with different power outputs. Table 1 presents the characteristics of the PV sources that were used in the simulations.

Table 1. PV Source Parameters.

Parameter	Photovoltaic Sources			
	PV1	**PV2**	**PV3**	**PV4**
V_{ocr} (V)	73.2	36.6	73.2	366
I_{scr} (A)	15.94	7.97	7.97	71.73
V_{mppr} (V)	58.6	29.3	58.6	293
I_{mppr} (A)	14.94	7.47	7.47	67.23
P_{mppr} (W)	876	219	438	19699
n_{iscT}	0.0010199	0.0010199	0.0010199	0.0010199
n_{vocT}	−0.00361	−0.00361	−0.00361	−0.00361

P_{mppr} is the maximum power that the PV source can deliver under STC. The rest of the parameters were presented and explained in Equations (6)–(10) at Section 2.1 and are used in the simulations to produce the current and voltage for different environmental conditions. PV2 of Table 1 is a commercial module and the rest of the systems are created by connecting multiple PV2 modules in series and/or in parallel. More specifically, PV1 is created by connecting two rows in parallel and each row consists of two modules connected in series. This will allow testing the MPPT methods for both higher current and voltage. Similarly, PV3 is formed by connecting two modules in series. This approach will test the methods for systems capable of producing higher voltage only. And lastly, PV4 represents a high-power source and is consisted of 9 rows connected in parallel, with each row composed by 10 modules connected in series.

Furthermore, testing the proposed methods under STC is necessary to validate their ability to achieve true MPP and, using additional values of temperature and irradiance, ensures that the control methods are capable to perform their task under varying environmental conditions.

Regarding DC-DC converter, the input and output capacitance values are chosen to be 1 mF, the inductance of the coil is set to 1 mH, the ohmic resistance of the diode is set to 1 mΩ and the switching frequency is set at 10 kHz. The load connected to the converter is 0.5 Ω and the duty cycle is restricted between 0.1 and 0.9.

2.5.2. Q-Learning and SARSA Implementation

The state vector is chosen to be (V, I, d), where V and I are the normalized and discretized values of voltage and current, respectively. The open circuit voltage and the short circuit current of each PV source under STC, are used for the normalization process, therefore normalized voltage and current values range between 0 and 1. The normalized values are then discretized to form a group of 20 possible value spaces that each variable can fall into. For example, when the output voltage of PV1 equals to 30 V, the normalized value would be $\frac{30}{73.2} \cong 0.41$ and the state variable V_{PV} after the discretization process would equal 8. The process is the same for finding the state variable I_{PV}. State variable d depends from the result of Equation (14):

$$deg = \tan^{-1}\left(\frac{dI_{PV}}{dV_{PV}}\right) + \tan^{-1}\left(\frac{I_{PV}}{V_{PV}}\right) \tag{14}$$

where V and I are the output voltage and current of the PV source at any moment, respectively. Variable deg in Equation (14) ranges between $-90°$ and $90°$, is used to separate between different environmental conditions and MPP is achieved only when it equals to 0. Therefore, state d is set to be 1 when deg $\in [-5°, 5°]$ and 0, in any other case. Consequently, the state space is formed by 800 states. Moreover, the action list is defined as A = {-0.1 -0.01 0 0.01 0.1} and each action is the change ΔD of the duty cycle sent to the PWM generator. In this manner, a Q-table is created, consisting of 800×5 state–action pairs.

As for the reward, it is important to discriminate between the three possible outcomes that an action can cause. For this reason, Equation (15) is used:

$$\mathrm{R} = \begin{cases} 1, \; deg \in [-5°, \; 5°] \; and \; \Delta D = 0 \\ \frac{\Delta P}{P_{mppr}}, \; otherwise \end{cases}. \tag{15}$$

This definition of the reward signal offers some advantages. First and foremost, the reward signal of 1, which is the higher possible value, is provided by the environment when the agent operates at MPP. In any other case the agent will receive a reward ranging from -1 to 1, since P_{mppr} represents the power produced under STC and ΔP is the power produced difference between two successive time steps. Incorporation of a negative reward encourages the agent to avoid actions with negative outcomes. Lastly, P_{mppr} is a fundamental PV characteristic given by all manufacturers and therefore, the same philosophy is used to define the reward for every PV source used in this paper.

Exploration is carried out by employing a modified version of the ε-greedy selection strategy. One of the main goals is to allow the agent to perform enough exploration actions when he's located on an unvisited or rarely visited state. For this reason, every state has a different corresponding ε value, initially set to 1, and that value is decreasing by 0.11 every time a random action is performed. This way, every time an agent visits the same state the probability of exploring will be decreasing from 1 to 0.89 to 0.78 and so on, until it reaches 0.01 which is the capped minimum value. The main strength of this modified ε-greedy selection strategy is that allows the agent to perform more likely exploration rounds when unvisited states, and less probably when frequently encountered states, are provided. Nevertheless, the agent will still perform random actions even after the MPP has been reached with probability 0.01 and power losses will occur, every now and then.

Figure 2 presents the flowchart of both algorithms. The update of Q(s, a) for Q-learning uses the update rule given in (11), while for SARSA Equation (12) is applied. The learning rate α is chosen to be 0.1 to allow convergence to the optimal policy, the discount factor γ is set to 0.9, to make the agent focus on long-term reward rather than the short-term, and the sampling time that simulations run for both algorithms is 0.01.

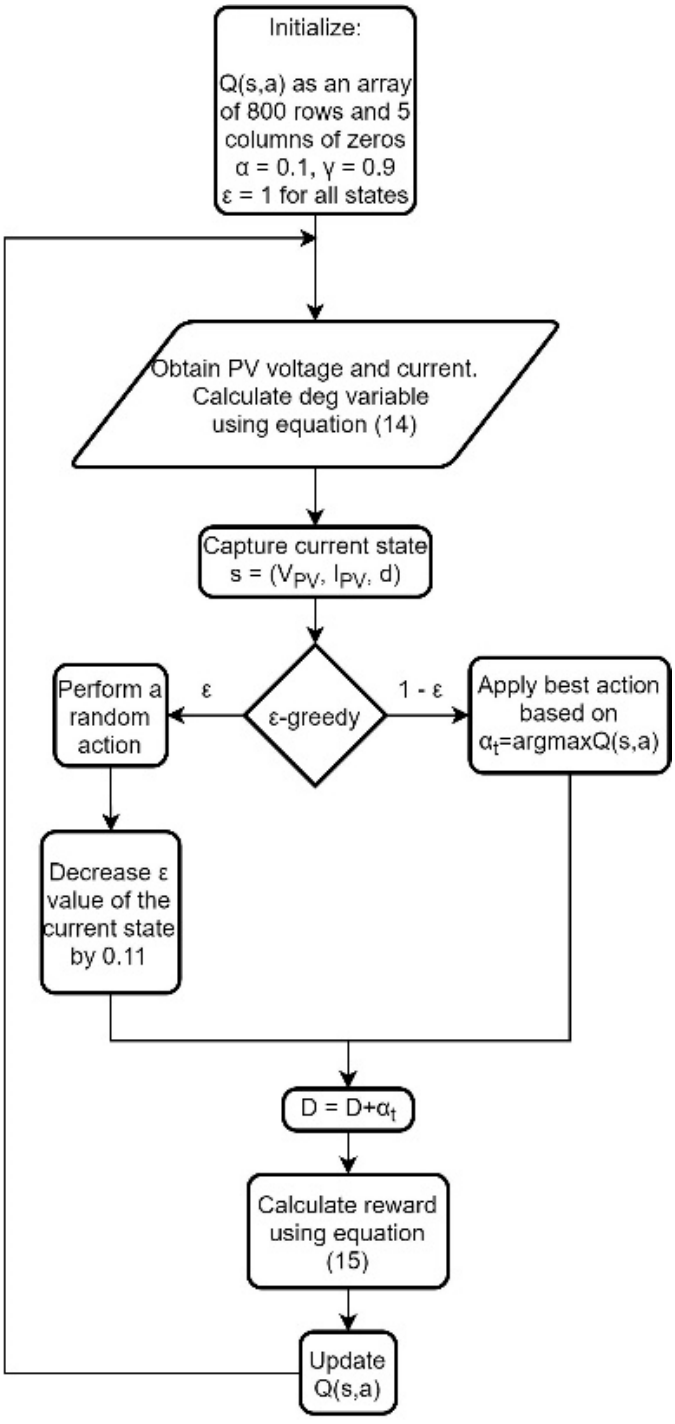


Figure 2. Q-learning/SARSA flowchart.

2.5.3. Evolutionary Algorithms

The chosen optimization variable is $\vec{x} = (a_1\ a_2\ a_4\ a_5\ a\ \gamma\ \varepsilon rr)$, where α_1, α_2, α_4, α_5 are the actions of the agent, previously set at −0.1, −0.01, 0.01, and 0.1, respectively, α is the learning rate and γ is the discount factor. It should be noted, that the action α_3, which equals zero, was not included due to the fact that oscillations after achieving MPP are undesirable. The variable called εrr stands for "ε reduction rate" and it is initially set to 0.11. By adjusting this variable, the agent will perform more or less exploration actions, depending on the value given. Both the initial population N and the number of iterations, are set to 30. The cost function used to evaluate the candidate solutions is given in Equation (16):

$$f = \frac{1}{1+E} \tag{16}$$

where E is the produced energy by the PV source over the 24-s simulation time.

Tables 2 and 3 present the search space for every parameter for BB—BC and for GA respectively.

Table 2. Search space of candidate solutions for BB—BC.

Parameter	Initial Value	Search Space	
		N − k + 1 Candidates	**K − 1 Candidates**
α_1	−0.1	[−0.15, −0.05]	$a_{1b} + 10r/k$
α_2	−0.01	[−0.05, −0.001]	$a_{2b} + 10r/k$
α_4	0.01	[0.001, 0.05]	$a_{4b} + 10r/k$
α_5	0.1	[0.05, 0.15]	$a_{5b} + 10r/k$
α	0.1	[0, 1]	$a_b + 10r/k$
γ	0.9	[0, 1]	$\gamma_b + 10r/k$
εrr	0.11	[0, 1]	$\varepsilon rr_b + 10r/k$

where r is a randomly created number following uniform distribution in space [−0.02, 0.02].

Table 3. Search space of candidate solutions for genetic algorithms (GA).

Parameter	Search Space	
	Upper Bound	**Lower Bound**
α_1	−0.05	−0.15
α_2	−0.001	−0.05
α_4	0.05	0.001
α_5	0.15	0.05
α	1	0
γ	1	0
εrr	1	0

The options of the GA algorithm are set according to Table 4.

Table 4. Option settings of GA.

Option	Value
Population Size	30
Number of Generation	30
Mutation Rate	0.01
Crossover Fraction	0.8
Selection Function	Roulette
Reproduction (EliteCount ceil)	0.05 × 30

3. Results

3.1. Q-Learning and SARSA for Different PV Sources and Environmental Conditions

The first stage of the simulations is the test of the proposed MPPT methods for different irradiance and temperature levels. These can be seen in Figure 3a,b, and represent frequently encountered operating conditions of most PV installation locations.

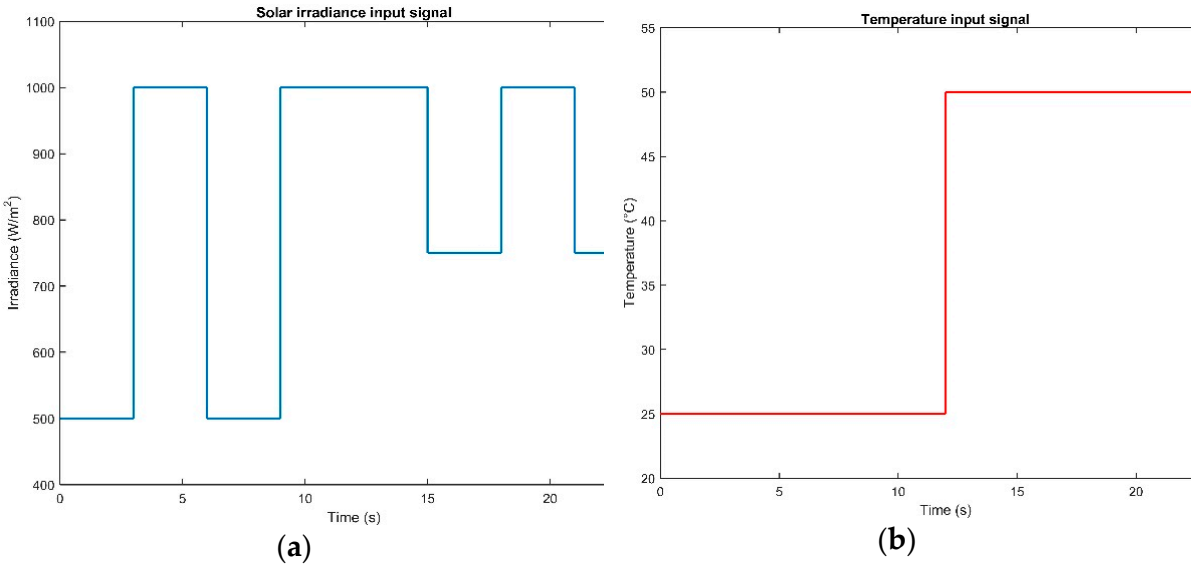


Figure 3. Simulation environmental conditions: (**a**) Incident irradiance on PV source; (**b**) PV temperature.

Figures 4a–7a show the power extraction at every moment of PV1, PV2, PV3, and PV4, respectively, for the Q-learning algorithm. The corresponding duty cycle of the buck converter can be seen in Figures 4b–7b. The Figures show that the agent performs exploration actions every time an unvisited state is provided. After enough exploration rounds have been completed, the agent is capable of reaching MPP very quickly once an explored state is given. For example, in Figure 4a the same environmental conditions are present between 0 and 3 s, and between 6 and 9 s. In the first case, almost half of the time the agent performs exploration actions, while in the latter, the agent exploits the knowledge gained from previous exploration rounds and selects the action with the best outcome. Additionally, it is worth noting that even though the MPP is not reached in some time periods, after the same environmental conditions are encountered, the agent exhibits better response by either achieving MPP or improving. Lastly, power losses periodically occur due to random actions, even after MPP has be achieved, due to the action selection strategy ε-greedy.

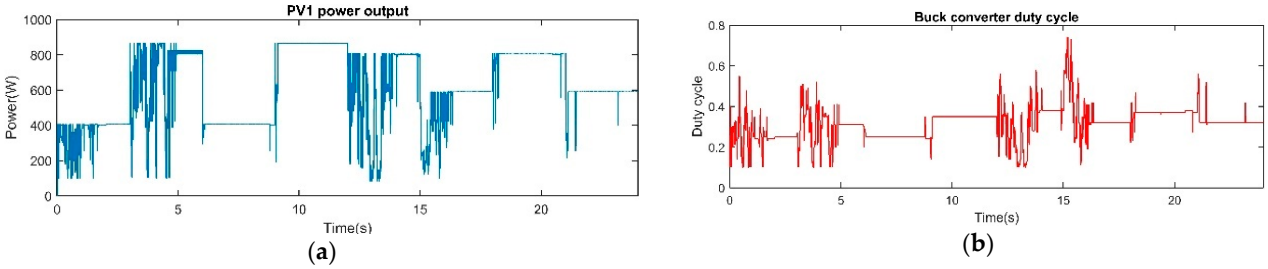
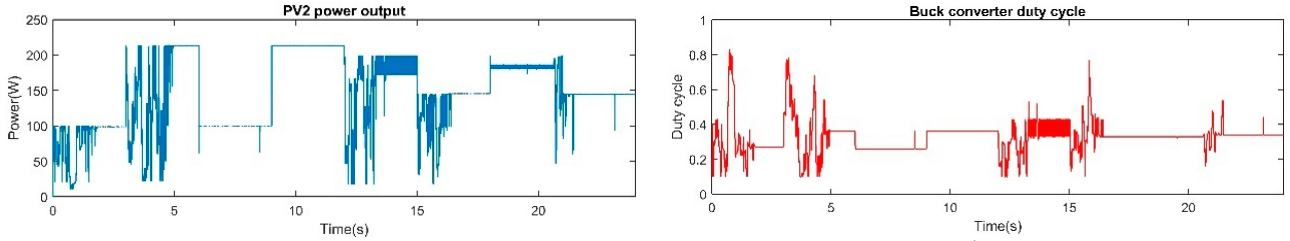
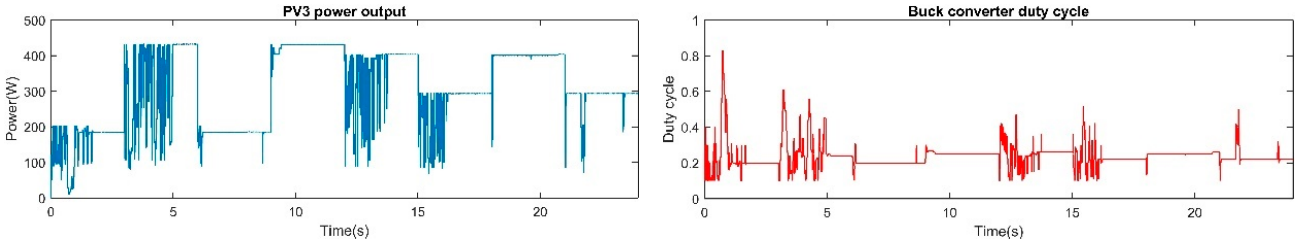


Figure 4. PV1 simulation results with Q-learning RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



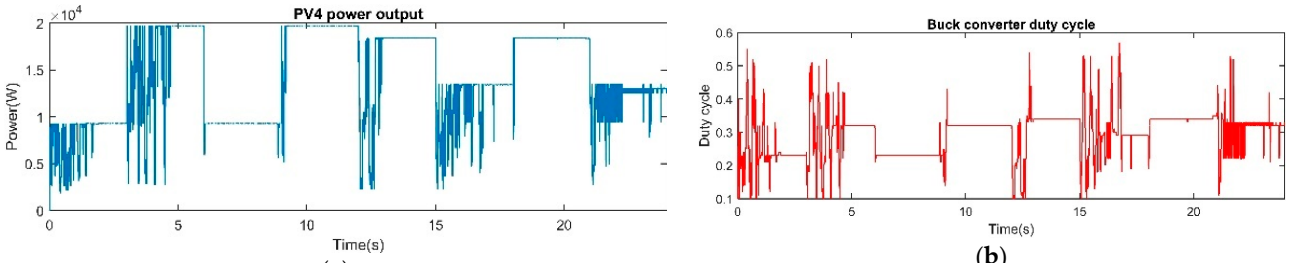
(**a**) (**b**)

Figure 5. PV2 simulation results with Q-learning RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

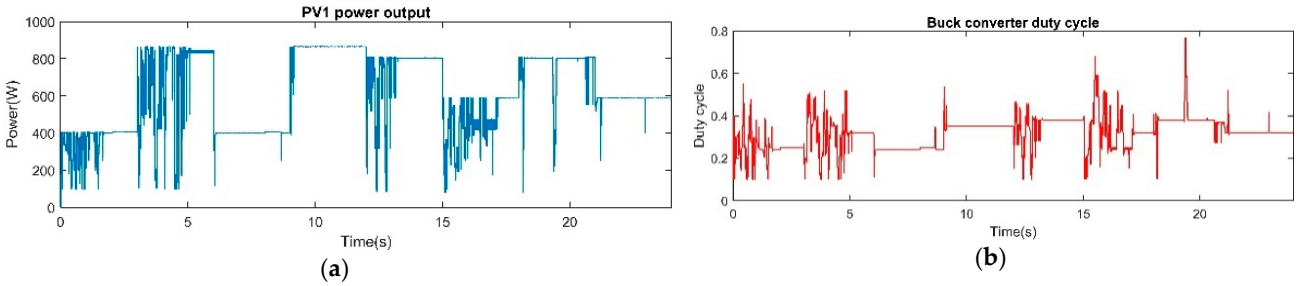
Figure 6. PV3 simulation results with Q-learning RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

Figure 7. PV4 simulation results with Q-learning RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

Figures 8a–11a show the produced power at every moment from PV1, PV2, PV3, and PV4, respectively, for the SARSA algorithm. The corresponding duty cycle of the buck converter is depicted in Figures 8b–11b. The performance of this method is similar to Q-learning with the only difference being that after the MPP has been found, this method stabilizes at it, in contrast with Q-learning that oscillations exist in some cases and can be seen in Figures 5a and 7a. It is also worth pointing out that both algorithms fail to find the exact MPP in time periods 0–3 s and 6–9 s for PV3. This can be seen in Figures 6a and 10a.



(**a**) (**b**)

Figure 8. PV1 simulation results with SARSA RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

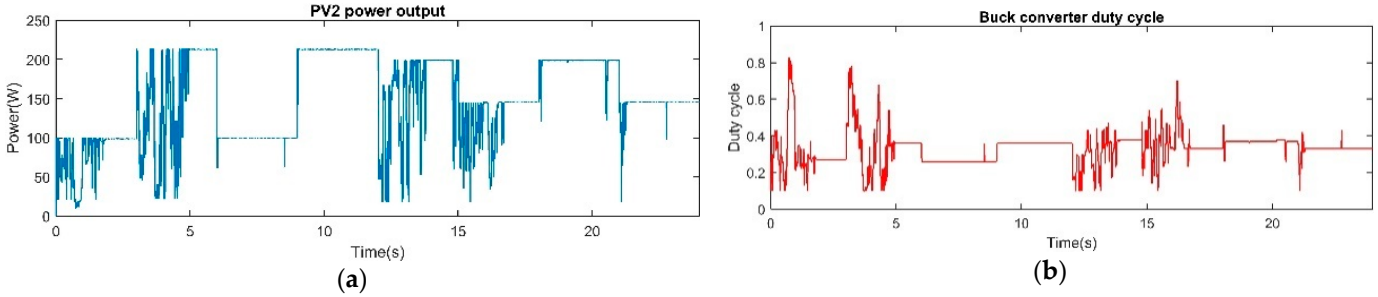


Figure 9. PV2 simulation results with SARSA RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

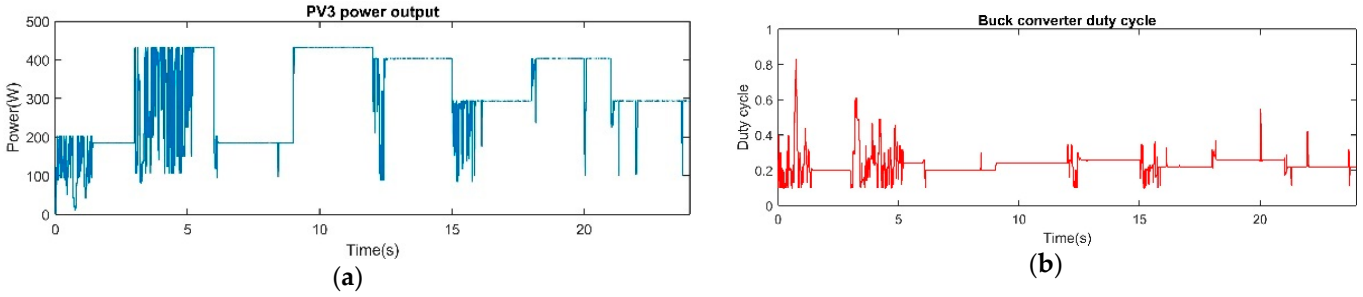


Figure 10. PV3 simulation results with SARSA RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

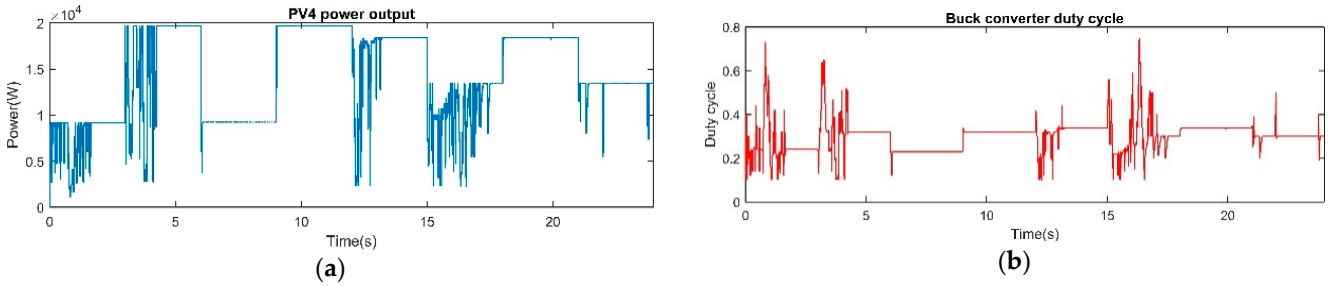


Figure 11. PV4 simulation results with SARSA RLMPPT method: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

Table 5 presents the produced energy for every PV source for both RL algorithms. The results suggest that SARSA is slightly better than Q-learning in terms of produced energy.

Table 5. Produced energy of different PV systems for each RL algorithm.

PV Source	Produced Energy (KJ)	
	Q-Learning Algorithm	SARSA Algorithm
PV1	14.489	14.700
PV2	3.476	3.522
PV3	7.216	7.369
PV4	332.880	338.590

3.2. Q-Learning and SARSA Optimization

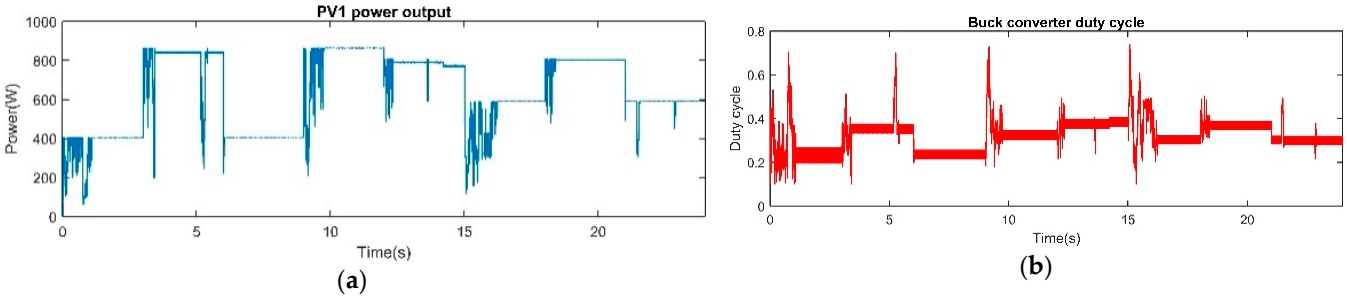
The second stage of the simulations is the application of BB—BC and GA for each algorithm on PV1 source and the validation of the results. Table 6 presents the parameter values obtained after applying BB—BC for each algorithm on PV1. These new parameter values are used in the simulations of the RLMPPT methods on the rest of the PV sources.

Table 6. Parameter values for each algorithm after the application of BB—BC.

Parameter	Initial Value	Value Obtained from BB—BC	
		Q-Learning Algorithm	SARSA Algorithm
α_1	−0.1	−0.0659	−0.0323
α_2	−0.01	−0.008	0.0196
α_4	0.01	0.0362	0.0281
α_5	0.1	0.1158	0.0424
α	0.1	0.4091	0.6071
γ	0.9	0.3430	0.0969
εrr	0.11	0.2491	0.2415

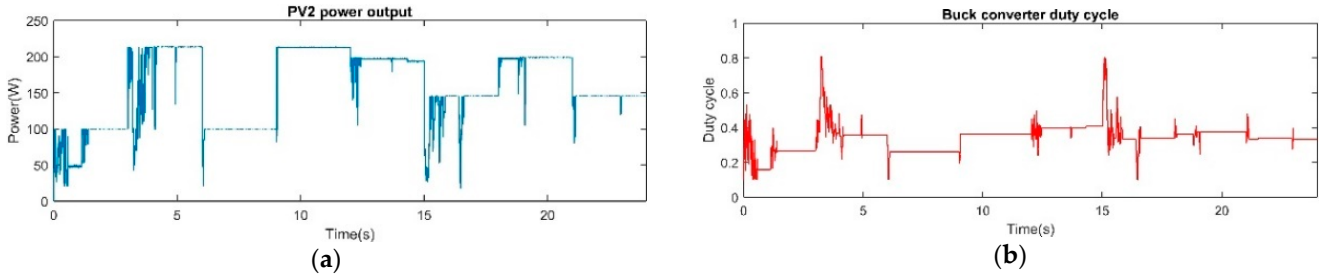
Because the initial population N and the number of iterations k, are set to 30, there is a probability that the optimal values are not found after the application of BB—BC. These values are selected after trials which aim to balance the running time of the algorithm in conjunction to obtained values that enhanced the overall performance. Higher values of N and k result to obtained values that lead in slightly better performance but the running time of the algorithm is increased exponentially. The optimization algorithm increased the εrr variable value for both algorithms, which leads to less exploration rounds performed by the agent. The discount factor was drastically reduced and the learning rate was increased, for both algorithms, leading to a more shortsighted agent and to faster convergence. Finally, after enough exploration rounds occur from every state, the agent will stop exploring completely. The reason for this change was to prevent power losses after finding the MPP.

Figures 12a–15a show the produced power every moment from each PV source for Q-learning algorithm with the optimized parameters. The corresponding duty cycle of the buck converter is given in Figures 12b–15b. Less exploration actions cause the agent to achieve operation at MPP much faster. Moreover, the Q-learning algorithm exhibited oscillations at certain time periods on PV2 and PV4, which are now eradicated.



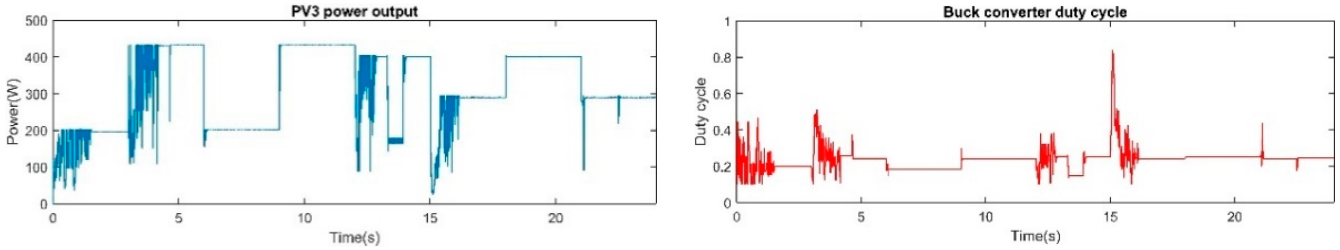
(**a**) (**b**)

Figure 12. PV1 simulation results with Q-learning RLMPPT method, after the optimization with BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



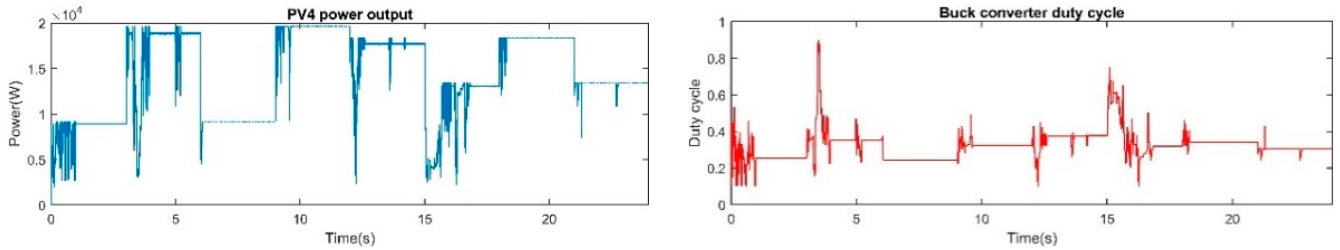
(**a**) (**b**)

Figure 13. PV2 simulation results with Q-learning RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

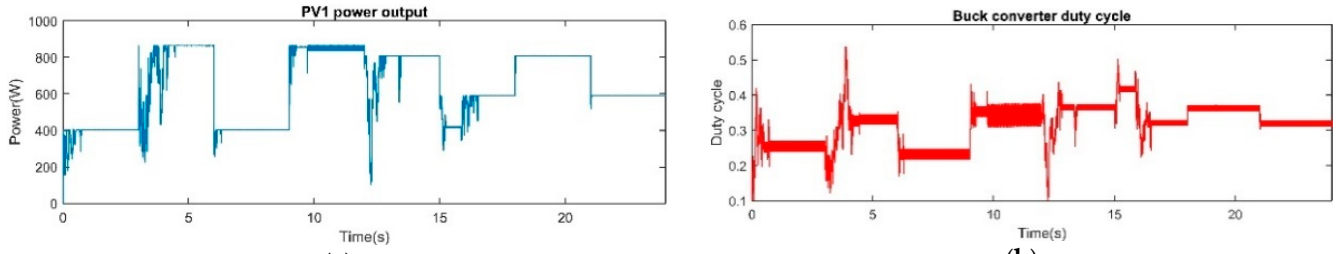
Figure 14. PV3 simulation results with Q-learning RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

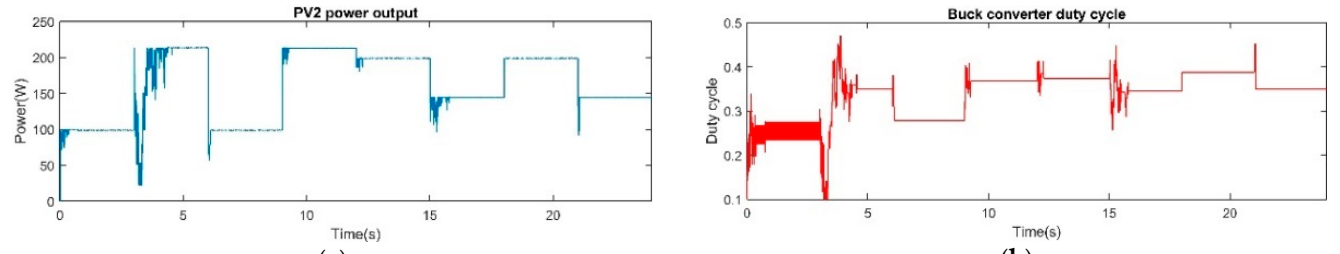
Figure 15. PV4 simulation results with Q-learning RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

The same graphs for the SARSA algorithm are presented in Figures 16–19. The exact MPP is achieved in most time periods and once previous environmental conditions occur, the agent can track the MPP rapidly. In addition, after achieving operation at the MPP, the control method does not exhibit oscillations around it, for the vast majority of time periods.



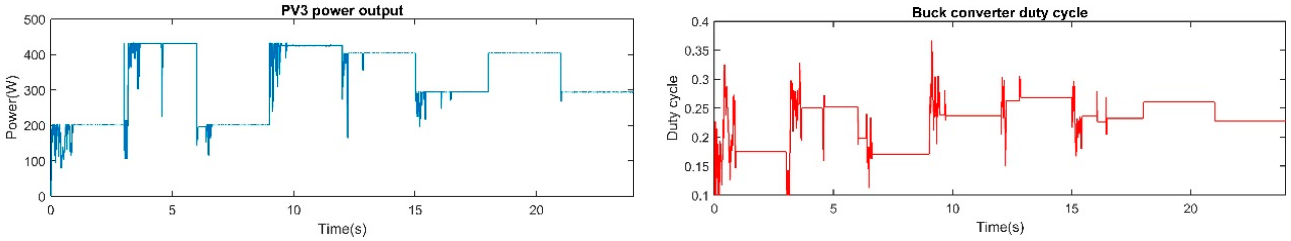
(**a**) (**b**)

Figure 16. PV1 simulation results with SARSA RLMPPT method, after the optimization with BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



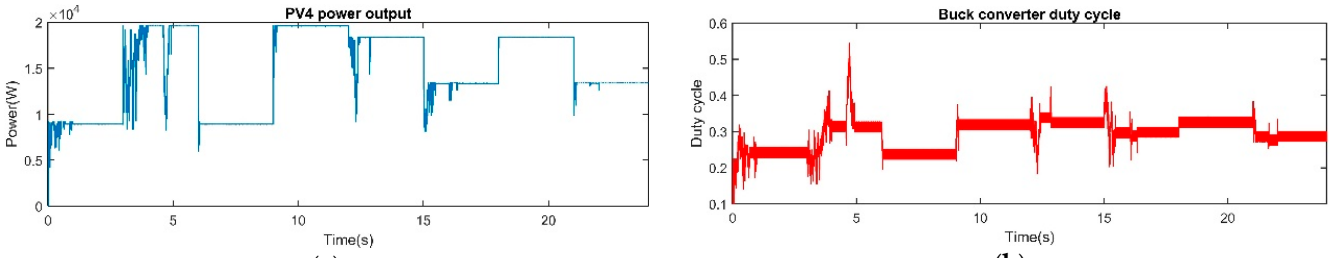
(**a**) (**b**)

Figure 17. PV2 simulation results with SARSA RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

Figure 18. PV3 simulation results with SARSA RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.



(**a**) (**b**)

Figure 19. PV4 simulation results with SARSA RLMPPT method, with usage of parameter values from BB—BC: (**a**) Output power of PV source; (**b**) buck converter duty cycle.

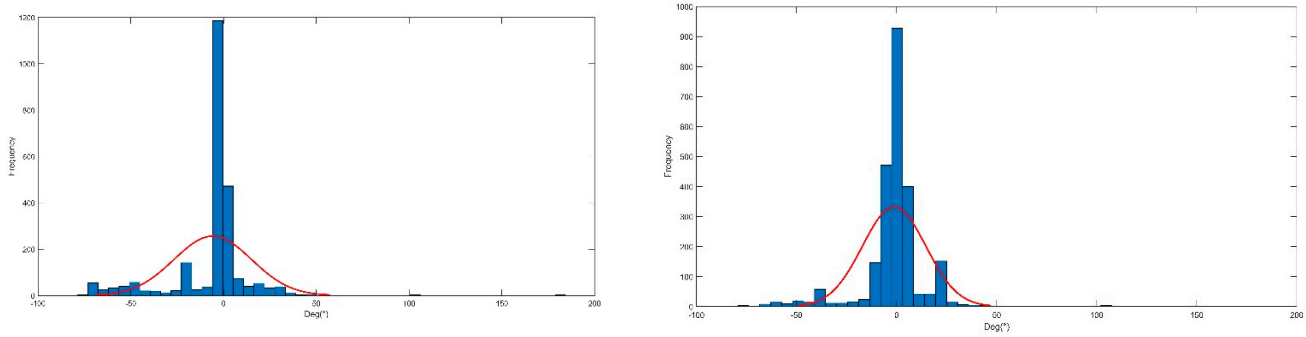
Table 7 presents the produced energy of each algorithm after the optimization of the aforementioned parameters, along with the percent change compared to the produced energy before the optimization by BB—BC.

Table 7. Produced energy for each algorithm and % change.

PV Source	Produced Energy(KJ)		% Change of Produced Energy	
	Q-Learning	SARSA	Q-Learning	SARSA
PV1	15.211	15.321	5.0	4.2
PV2	3.763	3.842	8.3	9.1
PV3	7.390	7.819	2.4	6.1
PV4	339.980	354.360	2.1	4.5

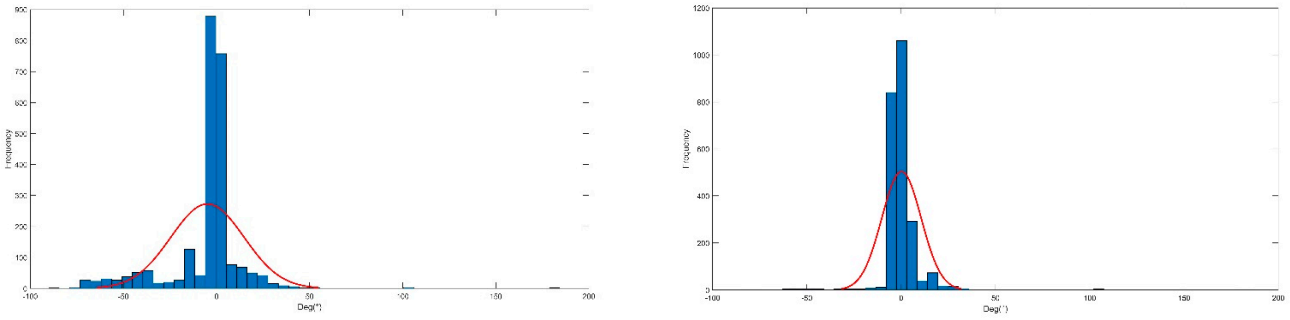
In Figures 20a and 21a the frequency distribution of variable deg for each RL algorithm, which results from Equation (14), can be seen. Figures 20b and 21b show that the optimization algorithm increases the frequency that variable deg is close to or equal to zero. This change is expected since the produced energy is higher after the application of BB—BC and PVs produce the most energy when variable deg is zero.

The parameters of the fitted Gaussian functions (center and standard deviation) for all the PV sources for both algorithms before and after the applications of the BB—BC algorithm are presented in Table 8. After the application of the BB—BC algorithm, in all cases the standard deviation is decreased and in most cases the center is moved closer to zero.



(**a**) (**b**)

Figure 20. Frequency distribution histogram for Q-learning algorithm of deg variable (**a**) before optimization and (**b**) after optimization with BB—BC.



(**a**) (**b**)

Figure 21. Frequency distribution histogram for SARSA algorithm of deg variable (**a**) before optimization and (**b**) after optimization with BB—BC.

Table 8. Parameters of Gaussian fitted functions.

PV Source	Parameters	Before BB—BC Application		After BB—BC Application	
		Q-Learning	SARSA	Q-Learning	SARSA
PV1	center	−5.82	−4.87	−1.23	0.15
	standard deviation	20.91	15.87	15.87	10.46
PV2	center	−6.54	−4.18	−6.70	0.73
	standard deviation	23.66	22.04	16.41	9.83
PV3	center	−3.63	−1.36	−1.24	2.71
	standard deviation	16.73	13.00	14.14	8.66
PV4	center	−5.78	−7.53	0.35	2.05
	standard deviation	18.63	20.90	14.15	9.73

In order to have a comparison measure for the performance of the optimization algorithm BB—BC, a genetic algorithm was applied to both RL methods. GAs are based on evolutionary theory and are capable of finding the globally optimum parameter according to Ref. [21]. Table 9 presents the values of the optimization variables and Table 10 exhibits the produced energy after the optimization with BB—BC and GA for each RL algorithm.

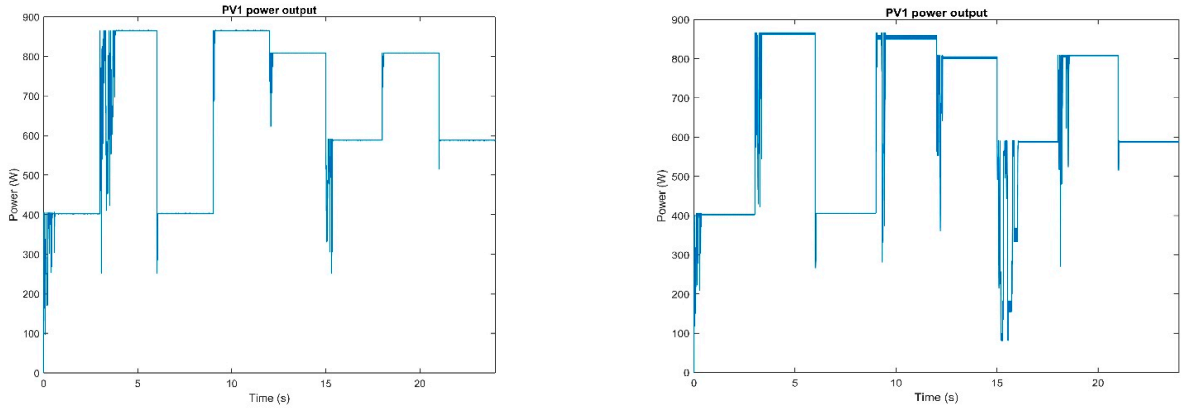
Table 9. Parameter values for each algorithm after the application of GA.

Parameter	Initial Value	Value Obtained from GA	
		Q-Learning Algorithm	SARSA Algorithm
α_1	−0.1	−0.0605	−0.0657
α_2	−0.01	−0.0110	−0.0045
α_4	0.01	0.0440	0.0252
α_5	0.1	0.1311	0.0542
α	0.1	0.8730	0.2609
γ	0.9	0.1112	0.0327
εrr	0.11	0.6552	0.2518

Table 10. Produced energy for each RL algorithm after the application of different optimization method.

PV Source	Produced Energy after BB—BC (KJ)		Produced Energy after GA (KJ)	
	Q-Learning	SARSA	Q-Learning	SARSA
PV1	15.211	15.321	15.504	15.740
PV2	3.763	3.842	3.836	3.540
PV3	7.390	7.819	7.613	7.454
PV4	339.980	354.360	331.050	350.960

Table 10 suggests that both optimization algorithms achieve similar energy results. GA is slightly better on the PV1 source that RL algorithms were optimized on. However, the SARSA algorithm which was optimized by BB—BC exhibits superior performance on PV2, PV3, and PV4. Figure 22a,b demonstrate the P-t graph for each RL algorithm after the application of GA.



(**a**) (**b**)

Figure 22. Output power of PV1 after optimization by GA: (**a**) With SARSA control method and (**b**) Q-learning method.

3.3. RL and FLC Comparison

The third stage of the simulations is the comparison of these methods with another soft computing method like a FLC. The FLC is a Takagi Sugeno Kang (TSK) controller with two inputs and one output. The first input of the controller is the slope β of the power–voltage (P–V) curve:

$$\beta(\rho) = \frac{P(\rho) - P(\rho - 1)}{V(\rho) - V(\rho - 1)} \tag{17}$$

where ρ is the sample time and equals to 0.01 sec (the same sample time is used in the proposed methods). At the MPP the slope ρ equals to zero. The second input is the change of the slope:

$$\Delta\beta(\rho) = E(\rho) - E(\rho - 1) \tag{18}$$

The output of the FLC is the change of the duty cycle ΔD which is accumulated to arise the duty cycle signal that drives the buck converter [30]. For both inputs, five triangular membership functions are used in the range of $[-1, 1]$. For this reason, the slope β and the $\Delta\beta$ need conditioning in the same range. The membership functions are presented in Figure 23. The scaling factors for conditioning the two input signals equals to 0.01 for the first input and to 0.1 for the second one. These values arise by trial-and-error method for the PV1 and the same procedure must be followed to obtained new values if another PV source is used.

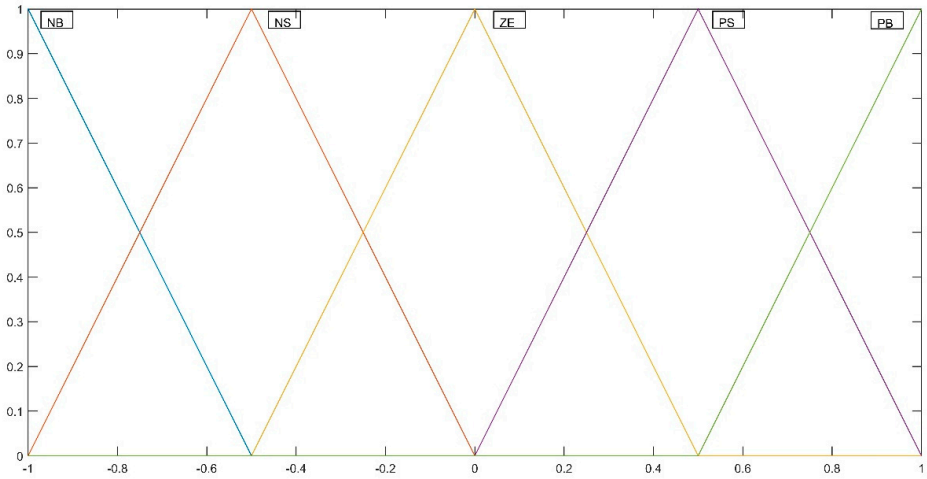


Figure 23. Membership functions of inputs.

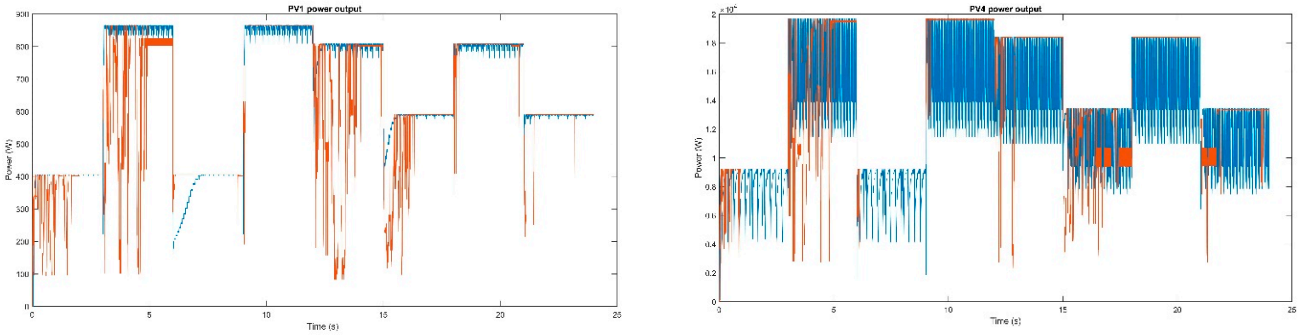
For the output, five singleton sets are used $\{PB = \frac{1}{0.1}, PS = \frac{1}{0.01}, ZE = \frac{1}{0}, NS = \frac{1}{-0.01}, NB = \frac{1}{-0.1}\}$. The PB, PS, ZE, NS, and NB stand for positive big, positive small, zero, negative small, and negative big, respectively. The fuzzy rules are presented in Table 11 and have been obtained by trials for the PV1.

Table 11. FLC rule base.

$\beta\backslash\Delta\beta$	PB	PS	ZE	NS	NB
PB	ZE	PS	PB	PB	PB
PS	NS	ZE	PS	PB	PB
ZE	PS	PS	ZE	NS	NS
NS	PS	ZE	NS	NB	NB
NB	ZE	NS	NB	NB	NB

The comparison between the FL controller and the Q-learning RLMPPT (before optimization) is presented in Figure 24a for the PV1 and in Figure 24b for the PV4. The Q-learning RLMPPT is selected for this comparison as its efficiency is a little bit lower than the SARSA. The PV1 and PV4 are selected for this comparison as they are two sources with completely different characteristics. In the PV1 the overall performance of the FLC is better than the Q-learning RLMPPT. For new environmental conditions, the Q-learning RLMPPT has to explore the new state–action pairs. On the other hand, for conditions that have been met before the performance of the Q-learning RLMPPT is better with no oscillations around the MPP. This is obvious in the time periods between 6–9 s, 9–12 s, 18–21 s, and 21–24 s. In the PV4, the overall performance of the Q-learning RLMPPT is better than the FLC. The FLC cannot successfully track the MPP and a lot of oscillations are observed. The FLC

needs resetting to its parameters for the specific source such as editing to the fuzzy rules and/or new values to the scaling factors of the inputs. This is a hard and time-consuming procedure. On the other hand, the Q-learning RLMPPT after the exploration phase is able to track the MPPT and for this source.



(**a**) (**b**)

Figure 24. Simulation results with Q-learning RLMPPT (red line) and FL controller (blue line): (**a**) Output power of PV1 source; (**b**) output power of PV4 source.

To conclude, the application of evolutionary optimization algorithms improves greatly the performance of both RL algorithms, which is illustrated in the P-t graphs. The optimized parameters allow the methods to find the exact MPP in time periods where they could not before the optimization and do so faster. For example, both methods could not find the exact MPP before optimization for PV3 and time periods 0–3 s and 6–9 s. Figures 14a and 18a indicate that, that is no longer the case. Furthermore, these parameters improve the energy production in every PV source, despite the fact that the evolutionary algorithms were applied for both algorithms only on PV1. Additionally, Table 5, Table 7, and Table 10 indicate that SARSA achieves higher energy production for every PV source, before and after the application of the optimization algorithms. Finally, the comparison between the Q-learning and the FLC highlights the universality of the proposed methods. The energy production of The FLC, for the PV1 source, equals to 15.664 KJ and is greater than the energy produced by Q-learning which equals to 14.488 KJ. On the other hand, the energy production of the FLC is much lower (295.566 KJ) compared to the energy produced by the Q-learning (335.942KJ) when a different source is used (PV4).

4. Discussion

In this paper, two universal RL methods are presented that can solve the MPPT problem by modeling it as an MDP. Both methods are independent of PV characteristics and can be applied on any PV array with the only knowledge required being V_{OC}, I_{SC}, and P_{MPP} under STC, which are always given by the manufacturer. Moreover, they display their MPP tracking ability for substantial changes of environmental conditions.

A fast and capable of converging to the optimal solution is achieved using evolutionary algorithms. These metaheuristic algorithms are also employed to further improve both proposed methods. The results suggest that, even though these algorithms are performed for each method on a particular PV source, the performance for both methods on different PV sources is remarkable. Furthermore, this study has shown that SARSA algorithm is superior to Q-learning in terms of energy production, since the energy extracted from every PV source is higher, with and without the parameter values from the optimization algorithm, than that of Q-learning algorithm.

Further research could be conducted to determine the effectiveness of the proposed methods under partial shading conditions. In addition, the exploration strategy could be modified in such way that actions already chosen before, would be unlikely to be reselected in contrast with actions never taken before. Additionally, since fourth decimal point of an action will barely change the performance of the RL methods, evolutionary

algorithms should focus on choosing candidate solutions belonging to a discrete set of values. In further study the proposed learning algorithms will be motivated for hardware implementation to real conditions.

Author Contributions: Methodology, K.B., A.D., and P.K.; software, A.D. and P.K.; validation, K.B., A.D. and P.K.; writing—original draft preparation, K.B., A.D. and P.K.; writing—review and editing, K.B., A.D., and P.K.; visualization, K.B. and P.K.; supervision, A.D.; project administration, A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Acronyms	
BB—BC	Big Bang—Big Crunch
FLC	Fuzzy Logic Controller
GA	Genetic Algorithm
IC	Incremental Conductance
MDP	Markov Decision Process
MPP	Maximum Power Point
MPPT	Maximum Power Point Tracking
NN	Neural Network
OCV	Open Circuit Voltage
PID	Proportional, Integral and Derivation
PSO	Particle Swarm Optimizer
PV	PhotoVoltaic
PWM	Pulse Width Modulation
P&O	Perturb & Observe
RES	Renewable Energy Sources
RL	Reinforcement Learning
SARSA	State Action Reward State Action
SCC	Short Circuit Current
STC	Standard Test Conditions
QT	Q Table
εrr	ε reduction rate
Variables	
D	Duty cycle
E	PV produced energy
E_{BBBC}	PV produced energy after optimization
f	Cost function
G_{PV}	Solar irradiance incident on PV
I_{PV}	Output current of PV
k	Iteration step
P_{in}	Input power of buck converter
P_{MPP}	PV power output at MPP
P_{out}	Output power of buck converter
R	Reward signal
r	Random normal number
Re_{PV}	Effective resistance of PV source
β	Slope of power voltage curve
s_t	Agent's current state
s_{t+1}	Agent's next state
T_{PV}	PV temperature
V_{in}	Input voltage of buck converter
V_{PV}	Output voltage of PV
V_{out}	Output voltage of buck converter
$\vec{x}^{c}$	Center of mass
$\vec{x}^{new}$	New candidate solutions
α	Learning rate
α_t	Agent's action
α_{t+1}	Agent's next action
γ	discount factor
$\Delta\beta$	Change of slope
ΔD	Duty cycle variation
ΔP	Power variation
Constants	
A	Action list
G_r	Reference irradiance incident on PV for STC
I_{MPP}	Current at MPP
I_{mppr}	PV current at MPP for STC
I_{SC}	PV short circuit current
I_{scr}	Short circuit current for STC
l	Optimization parameter upper limit
N	Initial population of solutions
n_{iscT}	Temperature coefficient of short circuit current
n_{vocT}	Temperature coefficient of open circuit voltage
P_{mppr}	PV power output at MPP for STC
T_r	Reference temperature for STC
V_{MPP}	Voltage at MPP
V_{mppr}	PV voltage at MPP for STC
V_{OC}	PV open circuit voltage
V_{ocr}	Open circuit voltage for STC
ρ	Sample time

References

1. Mao, M.; Cui, L.; Zhang, Q.; Guo, K.; Zhou, L.; Huang, H. Classification and summarization of solar photovoltaic MPPT techniques: A review based on traditional and intelligent control strategies. *Energy Rep.* **2020**, *6*, 1312–1327. [CrossRef]
2. Farhat, M.; Barambones, O.; Sbita, L. A Real-Time Implementation of Novel and Stable Variable Step Size MPPT. *Energies* **2020**, *13*, 4668. [CrossRef]
3. Macaulay, J.; Zhou, Z. A Fuzzy Logical-Based Variable Step Size P&O MPPT Algorithm for Photovoltaic System. *Energies* **2018**, *11*, 1340. [CrossRef]
4. Abdel-Salam, M.; El-Mohandes, M.-T.; Goda, M. An improved perturb-and-observe based MPPT method for PV systems under varying irradiation levels. *Sol. Energy* **2018**, *171*, 547–561. [CrossRef]
5. Ali Ahmed, I.-M.; Sayed Mahmoud, A.; Mohamed Essam, E.-M. Modified efficient perturb and observe maximum power point tracking technique for grid-tied PV system. *Int. J. Electr. Power Energy Syst.* **2018**, *99*, 192–202. [CrossRef]

6. Bounechba, H.; Bouzid, A.; Nabti, K.; Benalla, H. Comparison of Perturb & Observe and Fuzzy Logic in Maximum Power Point Tracker for PV Systems. *Energy Procedia* **2014**, *50*, 677–684. [CrossRef]
7. Feroz Mirza, A.; Mansoor, M.; Ling, Q.; Khan, M.I.; Aldossary, O.M. Advanced Variable Step Size Incremental Conductance MPPT for a Standalone PV System Utilizing a GA-Tuned PID Controller. *Energies* **2020**, *13*, 4153. [CrossRef]
8. Kumar, K.-K.; Bhaskar, R.; Koti, H. Implementation of MPPT Algorithm for Solar Photovoltaic Cell by Comparing Short-circuit Method and Incremental Conductance Method. *Procedia Technol.* **2014**, *12*, 705–715. [CrossRef]
9. Bouarroudj, N.; Boukhetala, D.; Feliu-Batlle, V.; Boudjema, F.; Benlahbib, B.; Batoun, B. Maximum Power Point Tracker Based on Fuzzy Adaptive Radial Basis Function Neural Network for PV-System. *Energies* **2019**, *12*, 2827. [CrossRef]
10. Kofinas, P.; Dounis, A.-I.; Papadakis, G.; Assimakopoulos, M.N. An Intelligent MPPT controller based on direct neural control for partially shaded PV system. *Energy Build.* **2015**, *90*, 51–64. [CrossRef]
11. Dounis, A.-I.; Kofinas, P.; Papadakis, G.; Alafodimos, C. A direct adaptive neural control for maximum power point tracking of photovoltaic system. *Sol. Energy* **2015**, *115*, 145–165. [CrossRef]
12. Dounis, A.-I.; Kofinas, P.; Alafodimos, C.; Tseles, D. Dynamic Neural Control for Maximum Power Point Tracking of PV system. In Proceedings of the 11th Symposium on Neural Network Applications in Electrical Engineering, Belgrade, Serbia, 20–22 September 2012; pp. 253–257. [CrossRef]
13. Hassan, T.-U.; Abbassi, R.; Jerbi, H.; Mehmood, K.; Tahir, M.F.; Cheema, K.M.; Elavarasan, R.M.; Ali, F.; Khan, I.A. A Novel Algorithm for MPPT of an Isolated PV System Using Push Pull Converter with Fuzzy Logic Controller. *Energies* **2020**, *13*, 4007. [CrossRef]
14. Liu, C.-L.; Chen, J.-H.; Liu, Y.-H.; Yang, Z.-Z. An Asymmetrical Fuzzy-Logic-Control-Based MPPT Algorithm for Photovoltaic Systems. *Energies* **2014**, *7*, 2177–2193. [CrossRef]
15. Kececioglu, O.F.; Gani, A.; Sekkeli, M. Design and Hardware Implementation Based on Hybrid Structure for MPPT of PV System Using an Interval Type-2 TSK Fuzzy Logic Controller. *Energies* **2020**, *13*, 1842. [CrossRef]
16. Dounis, A.-I.; Stavrinidis, S.; Kofinas, P.; Tseles, D. Fuzzy-PID controller for MPPT of PV system optimized by Big Bang-Big Crunch algorithm. In Proceedings of the International Conference on Fuzzy Systems (FUZZ-IEEE), Constantinople, Turkey, 2–5 August 2015; pp. 1–8. [CrossRef]
17. Dounis, A.-I.; Kofinas, P.; Alafodimos, C.; Tseles, D. Adaptive fuzzy gain scheduling PID controller for maximum power point tracking of photovoltaic system. *Renew. Energy* **2013**, *60*, 202–214. [CrossRef]
18. Kumar, A.; Kumar, A.; Arora, R. Overview of Genetic Algorithm Technique for maximum Power Point Tracking (MPPT) of Solar PV System. *IJCA Int. J. Comput. Appl.* **2015**, *3*, 21–24.
19. Hadji, S.; Gaubert, J.-P.; Krim, F. Real-Time Genetic Algorithms-Based MPPT: Study and Comparison (Theoretical an Experimental) with Conventional Methods. *Energies* **2018**, *11*, 459. [CrossRef]
20. Amirjamshidi, Z.; Mokhtari, Z.; Moussavi, Z.; Amiri, P. MPPT Controller Design Using Fuzzy-BBBC Method. *ACSIJ Adv. Comput. Sci. Int. J.* **2015**, *4*, 14:1–14:6.
21. Erol, O.-K.; Eksin, I. A new optimization method: Big Bang-Big Crunch. *Adv. Eng. Softw.* **2006**, *37*, 106–111. [CrossRef]
22. Kofinas, P.; Doltsinis, S.; Dounis, A.-I.; Vouros, G.-A. A reinforcement learning approach for MPPT control method of photovoltaic sources. *Renew. Energy* **2017**, *108*, 461–473. [CrossRef]
23. Hsu, R.-C.; Liu, C.-T.; Chen, W.-Y.; Hsieh, H.-I.; Wang, H.-L. A Reinforcement Learning-Based Maximum Power Point Tracking Method for Photovoltaic Array. *Int. J. Photoenergy* **2015**, *2015*, 496401. [CrossRef]
24. Chou, K.-Y.; Yang, S.-T.; Chen, Y.-P. Maximum Power Point Tracking of Photovoltaic System Based on Reinforcement Learning. *Sensors* **2019**, *19*, 5054. [CrossRef] [PubMed]
25. Yousef, A.; El-Telbany, M.-E.S.; Zekry, A. Reinforcement Learning for Online Maximum Power Point Tracking Control. *J. Clean Energy Technol.* **2015**, *4*, 245–248. [CrossRef]
26. Rühle, S. Tabulated values of the Shockley–Queisser limit for single junction solar cells. *Sol. Energy* **2016**, *130*, 139–147. [CrossRef]
27. Coelho, R.F.; Concer, F.; Martins, D.C. A study of the basic DC-DC converters applied in maximum power point tracking. In Proceedings of the 2009 Brazilian Power Electronics Conference, Bonito-Mato Grosso do Sul, Brazil, 27 September–1 October 2009; pp. 673–678.
28. Sutton, R.-S.; Barto, A.-G. References. In *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2018; ISBN 9780262039246.
29. Kaelbling, L.-P.; Littman, M.-L.; Moore, A.-W. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
30. Robles Algarín, C.; Taborda Giraldo, J.; Rodríguez Álvarez, O. Fuzzy Logic Based MPPT Controller for a PV System. *Energies* **2017**, *10*, 2036. [CrossRef]