Article

A Novel Real-Time PV Error Handling Exploiting Evolutionary-Based Optimization

Asimina Dimara [1,2,*], **Alexios Papaioannou** [1,3,*], **Konstantinos Grigoropoulos** [1], **Dimitris Triantafyllidis** [1], **Ioannis Tzitzios** [1], **Christos-Nikolaos Anagnostopoulos** [2], **Stelios Krinidis** [1,3], **Dimosthenis Ioannidis** [1] and **Dimitrios Tzovaras** [1]

1   Centre for Research and Technology Hellas, Information Technologies Institute, 57001 Thessaloniki, Greece; grigokos@iti.gr (K.G.); dimitrios.triantaf@gmail.com (D.T.); tzijonnis@gmail.com (I.T.); krinidis@iti.gr (S.K.); djoannid@iti.gr (D.I.); dimitrios.tzovaras@iti.gr (D.T.)
2   Department of Cultural Technology and Communication, Intelligent Systems Lab, University of the Aegean, 81100 Mytilene, Greece; canag@aegean.gr
3   Management Science and Technology Department, International Hellenic University (IHU), 65404 Kavala, Greece
*   Correspondence: adimara@iti.gr (A.D.); alexiopa@iti.gr (A.P.)

Abstract: The crucial need for perpetual monitoring of photovoltaic (PV) systems, particularly in remote areas where routine inspections are challenging, is of major importance. This paper introduces an advanced approach to optimizing the maximum power point while ensuring real-time PV error handling. The overarching problem of securing continuous monitoring of photovoltaic systems is highlighted, emphasizing the need for reliable performance, especially in remote and inaccessible locations. The proposed methodology employs an innovative genetic algorithm (GA) designed to optimize the maximum power point of photovoltaic systems. This approach takes into account critical PV parameters and constraints. The single-diode PV modeling process, based on environmental variables like outdoor temperature, illuminance, and irradiance, plays a pivotal role in the optimization process. To specifically address the challenge of perpetual monitoring, the paper introduces a technique for handling PV errors in real time using evolutionary-based optimization. The genetic algorithm is utilized to estimate the maximum power point, with the PV voltage and current calculated on the basis of simulated values. A meticulous comparison between the expected electrical output and the actual photovoltaic data is conducted to identify potential errors in the photovoltaic system. A user interface provides a dynamic display of the PV system's real-time status, generating alerts when abnormal PV values are detected. Rigorous testing under real-world conditions, incorporating PV-monitored values and outdoor environmental parameters, demonstrates the remarkable accuracy of the genetic algorithm, surpassing 98% in predicting PV current, voltage, and power. This establishes the proposed algorithm as a potent solution for ensuring the perpetual and secure monitoring of PV systems, particularly in remote and challenging environments.

Keywords: photovoltaic parameters; PV error handling; evolutionary algorithm; PV monitoring

1. Introduction

Photovoltaic (PV) systems are a remarkable breakthrough that dazzle with their unparalleled benefits. The energy harnessed by these systems is devoid of any pernicious pollutants and does not impinge on natural resources or threaten the well-being of living beings. In addition, the noiseless and highly reliable PVs are entirely self-sufficient, requiring only nominal maintenance costs [1]. It is no wonder that solar PVs are considered a preeminent source of clean energy, and their prevalence continues to rise because of their groundbreaking advances in both technology and marketability. These strides include an impressive array of cell types, PV efficiency, and electronic capabilities that elevate their prominence and secure their position as a pinnacle of modern technology [2].

Although photovoltaic systems offer a multitude of benefits, it should be noted that their output is subject to fluctuations that are dependent on several factors, including prevailing outdoor conditions such as solar irradiance and temperature, as well as the characteristics of the PV modules, such as photovoltaic cell temperature and connected load [3]. To ensure optimal performance, the connected load must operate at a specific juncture, known as the maximum power point (MPP), where the PV produces the maximum power output. This feat is accomplished by accurately correlating the PV's characteristics with the environmental variables to achieve maximum point power tracking (MPPT).

The exhaustive testing of all commercially available PV models under different environmental conditions is an arduous and cost-intensive task. Consequently, researchers have exploited mathematical and physical models to simulate, estimate, and forecast the PV's electrical characteristics based on diverse parameters [4]. The most commonly used models are lumped parameter models with varying numbers of diodes. The single-diode model of the PV cell reigns supreme as the most prevalent model employed today [5]. However, determining the current–voltage correlation of the single-diode photovoltaic is an intricate process that involves solving complex transcendental and nonlinear equations, rendering it a laborious undertaking.

Establishing all pertinent parameters is imperative to accurately model a single-diode photovoltaic model [6]. However, not all parameters and specifications are readily available in the manufacturer's manual. While data regarding voltage at MPP (V_{mpp}), open circuit voltage (V_{oc}), current at MPP (I_{mpp}), and short circuit current (I_{sc}) are commonly provided, other critical PV parameters, such as diode ideality factor (*a*), parallel resistance (R_p), reverse saturation current (I_s), and photogenerated current (I_{ph}), may not be easily obtained [6]. Consequently, estimation or modeling using alternative means becomes necessary to determine these unknown parameters. Given the complexity of the process, accurately modeling PV characteristics is essential to ensuring model robustness and reliability.

Estimating photovoltaic parameters is a multifaceted endeavor that relies on diverse numerical, statistical, analytical, and intelligent techniques. Although numerical models rely solely on manufacturer-provided data to estimate PV parameters, analytical and statistical approaches utilize complex non-linear equations to estimate the I–V curve parameters [7]. Intelligent models leverage meta-heuristic algorithms to extract PV parameters, making use of sophisticated techniques to tackle the challenge of PV modeling [8]. Indeed, the extensive literature [9] on extracting PV parameters attests to the formidable nature of this task.

From the perspective of PV estimation, the paramount and pivotal factors are PV modeling and its parameters, as well as the MPPT. In addition, it is crucial to ensure the uninterrupted and continuous functioning of photovoltaics. Integrating all PV parameters and methods to estimate the I–V curve, while also providing techniques for error handling, is a challenging task. The complexity further increases when using fusion and hybrid methods to ensure that the PV modeling output closely approximates the actual PV output, which is essential for optimizing the power process and, subsequently, error handling.

In light of the challenges and concerns mentioned above, this paper proposes a novel approach to address the modeling and optimization of single-diode photovoltaic systems. Specifically, an analytical method is employed to model the photovoltaic system, while a state-of-the-art genetic algorithm is utilized to estimate the electric parameters of the system. The resulting real-time values of V_{act}, I_{act}, and P_{act} are then compared to the predicted values obtained from the genetic algorithm (V_{gen}, I_{gen}, and P_{gen}), which provides a means of identifying potential malfunctions or errors in the PV system or its sensing devices. By notifying the end-user of any significant deviations between the predicted and actual values, appropriate actions can be taken to prevent further issues and ensure the smooth and continuous operation of the photovoltaic system [10]. Overall, this approach holds great promise for addressing the challenges associated with accurate PV modeling

and error handling in single-diode photovoltaic systems while eliminating the integration of complex IoT systems [11].

Recent studies and advancements in PV modeling have indicated that, for the specific task of estimating the I–V curve and optimizing the power process, the inclusion of certain parameters (e.g., air humidity, wind direction, wind speed) may not yield a significant improvement in accuracy [12]. The selected PV modeling method, while simplified in its consideration of operating conditions, has been validated against real-world data, demonstrating high accuracy and reliability [12]. Moreover, the decision to limit the number of parameters considered in the model is also pragmatic. By focusing on essential factors, the complexity of the model is reduced while minimizing associated costs. Additional parameters would require more sophisticated sensing devices and potentially lead to increased expenses in both installation and maintenance for real-time conditions.

Within this context, the proposed method holds profound significance in the field of PV systems by introducing a novel approach to real-time monitoring and error detection. The fundamental concept revolves around the premise that, under optimal conditions, the PV modeling outputs, which estimate the electricity generation (represented by V_{gen}, I_{gen}, and P_{gen}), should closely align with the actual measurements of the PV system's performance (V_{act}, I_{act}, and P_{act}). Any substantial deviation between these two sets of values indicates the potential occurrence of errors or malfunctions within the PV system or its sensing devices. Exploiting a GA for parameter estimation, the method enables accurate and low-cost real-time monitoring of PV operations. This not only ensures the flawless functionality of the PV system but also provides a proactive mechanism for identifying and addressing issues promptly, thereby maximizing the overall efficiency and reliability of the photovoltaic installation.

The contributions and novelty of this paper can be summarized as follows:

- Development of a novel genetic algorithm for MPP optimization in single-diode photovoltaic systems.
- Integration of the Lambert function into the optimization function of the genetic algorithm to ensure PV constraints are met.
- Implementation of a real-time PV error detection system to detect and notify the end-user of any potential malfunction or deviation between real and predicted PV values.

In the subsequent sections of this paper, a comprehensive analysis of photovoltaic (PV) output power estimation and fault diagnosis is presented. Section 2 addresses the extensive research conducted in this field, highlighting its significance in the renewable energy domain. In Section 3, a detailed examination of the various methodologies utilized for PV modeling, MPP optimization, and PV error handling is provided. The ensuing section, Section 4, showcases the outcomes of the aforementioned methodologies. Finally, the paper ends with a discussion and conclusive section encapsulating the key takeaways and insights gleaned from the research conducted.

2. Related Work

A model that can estimate, simulate, or predict a single-diode PV's energy production must be accurate and efficient while monitoring the MPPT. PV models consist of non-linear characteristics between current (I) and voltage (V). It is of ultimate importance to minimize errors when estimating the I–V curve. The higher the estimation, the better the model. Notwithstanding, even if the PV modeling is accurate, there are many other errors that might occur with the PV systems. The PV system fails due to abnormal module power operation, sensor problems, and other issues. Tracking those errors and reporting them to stakeholders is crucial for a continuous PV operation.

As presented in the work of [13], to accurately predict the output of a PV, the estimation must be associated with outdoor environmental parameters. Weather data were periodically retrieved (i.e., every 5 min) for two years. The selected outdoor values monitored are ambient temperature (*Ta*), relative humidity (*RH*), wind speed (*WS*), and global horizontal irradiance (*GHI*). Moreover, the respective PV parameters were monitored for the same

time interval and period of time and are the maximum PV power output (*Pmax*), the temperature of the PV modules (*Tm*), and the plane of array irradiance (*POA*). *Pmax* was correlated with all the weather variables to indicate their association. The lowest error was produced when the model consisted of *RH*, *WS*, and *Ta*. Therefore, it is suggested to forecast PV power using those variables.

According to [14], the MPPT may be estimated through the curve of the solar array by comparing power (P) and static conductance (G). Their model generates fluctuating power and is compared to the PV's array of actual instantaneous power. The error produced is used to modify the static conductance while reaching the MPP. Moreover, the generated error is applied to change the power reference until it attains a limit cycle near the MPP. Their model ensures that the MPPT operates accurately between 10% and 100% nominal power. As stated in [15], to estimate the PV parameters for better PV accuracy, the proper circuit model must be indicated. A flexible particle swarm optimization (FPSO) algorithm is exploited. The FPSO's ordinary movement is altered to boost its accuracy. The FPSO is compared to many other widely used algorithms, like the bird mating optimizer and particle swarm optimization. As stated above, the non-linear characteristics of the PV cells result in a continuous fluctuation of their values. Consequently, the proper modeling of the PV and its parameters is of key importance.

As addressed in the work of [16], it is crucial that the PV's operations be continuous without faults or disruptions. The MPPT enhances the efficiency of the modules through DC–DC power optimizers. However, there are still many errors that might occur to the PV systems, like single component failure and plants out of service. As a result, PV auditing and monitoring are essential for diagnostic and maintenance purposes. PV monitoring is deployed through a day-ahead PV output prediction while faults are diagnosed. Moreover, a thorough examination is applied to the PV forecast through its performance indexes. In [17], it is reported that the I–V and P–V parameters and characteristics of a single-diode model affect the PV's operation. The authors divide the voltage and PV current output through an intermediate parameter. They simulate the PV's operation and analyze how each parameter affects the PV output and the error degree. The decomposition of voltage and current in the I–V simplifies the estimation of PV parameters while demonstrating the effect on MPP.

As far as monitoring PV techniques are concerned in [18], a novel method for smoothing photovoltaic power fluctuations using a hybrid storage system and optimized charging/discharging cycles is presented. The approach incorporates machine learning-based PV failure detection, reducing supercapacitor operation and computation time compared to traditional methods, as validated in experiments. In [19], the application of remote sensing (RS) technology in various stages of solar PV system development is explored, including potential estimation, array detection, fault monitoring, and cross-cutting areas. The comprehensive review highlights RS as a crucial tool for effective resource assessment, data analysis, and health monitoring, offering valuable support for PV system planning, management, and decision-making. Finally, in [20], a real-time fault detection and performance monitoring system for PV systems is presented. Using various sensors, the system identifies issues such as encapsulation failure and module corrosion. It employs a one-diode model, Python software, and a database generated from field testing to detect faults, providing users with easy-to-understand error messages and remote access for efficient PV system management and maintenance.

Based on all the above, it may be observed that, while significant progress has been made in modeling and monitoring single-diode PV systems, there remains a notable research gap concerning the integration of accurate parameter estimation, real-time error detection, and efficient MPPT mechanisms. The existing literature has predominantly focused on individual aspects, such as parameter estimation techniques, environmental parameter correlations, and fault detection systems. However, there is a need for a holistic approach that combines these elements to enhance the robustness and reliability of PV models. This paper aims to bridge this research gap by proposing a comprehensive

methodology. The objectives include the development of a novel genetic algorithm for accurate MPPT optimization, the integration of the Lambert function to ensure adherence to PV constraints, and the implementation of a real-time error detection system. The proposed approach seeks to address the challenges associated with accurate modeling and error handling in single-diode PV systems, offering a more complete and effective solution for continuous and fault-free PV operation.

3. Methodology

3.1. Single-Diode PV Modeling

A PV module consists of a number of PV cells connected in series to increase the voltage output and a number of PV cells connected in parallel to increase the current output. PV modules can generate electricity autonomously or combine with more PV modules in series or in parallel to form a PV array that generates a greater amount of electricity. The equivalent circuit of a PV module can be observed in Figure 1.
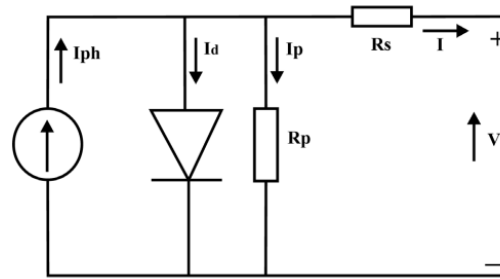

Figure 1. PV module equivalent circuit.

The output current of a PV module can be calculated by the following calculation using Kirchhoff's first law [21]:

$$I = N_p I_{ph} - N_p I_d - I_p,\qquad(1)$$

where I is the output current of the PV module, N_p is the number of cells connected in parallel, I_{ph} is the photogenerated current, I_p is the current flowing in the parallel resistance of the PV module R_p, and I_d is the diode's current, as described by Shockley [22]:

$$I_d = I_s \left[e^{\left(\frac{V_d}{nV_t}\right)} - 1 \right],\qquad(2)$$

where I_s is the diode's saturation current, V_d is the diode's voltage, n is the diode's ideality factor, and V_t is the diode's thermal voltage, which is calculated as follows:

$$V_t = \frac{kTN_s}{q_e},\qquad(3)$$

where k is the Boltzmann constant [23], T (in Kelvin) is the diode's temperature, N_s is the number of PV cells connected in series, and q_e is the elementary charge. Combining (1) with (2) and (3), the output current of the photovoltaic module is:

$$I = N_p I_{ph} - N_p I_s \left[e^{\left(\frac{q_e(V+IR_s)}{nkTN_s}\right)} - 1 \right] - \frac{V + IR_s}{R_p},\qquad(4)$$

where V is the output voltage of the PV module and R_s is the series resistance of the PV module.

Combining more than one PV module, a PV array can be formed, which constitutes the electrical power generator unit of a PV power station. The way that PV modules are connected defines the output voltage and the output current of the PV array. If there are n PV modules connected in series and m PV modules connected in parallel, then the

output voltage and the output current of the PV array are given, respectively, by the following equations.

$$V_{array} = nV, \tag{5}$$

$$I_{array} = mI, \tag{6}$$

where V and I are the output voltage and current, respectively, of a single PV module. The number of parameters used in this model are summarized in Table 1.

Table 1. PV modeling parameters.

Parameters	Description	Unit	Parameter Range
N_p	No. of cells connected in parallel in a PV module	unitless	>0
N_s	No. of cells connected in series in a PV module	unitless	>0
I_{ph}	Photo-generated current	A	0 to saturation current
I_s	Diode's saturation current	A	10^{-14}–10^{-17} A/μm^2
I_d	Diode's current	A	-10^{-9} to 2
V_t	Diode's thermal voltage	V	2 mV–30 mV
R_p	Parallel resistance	Ohm	0 to smallest resistance
R_s	Series resistance	Ohm	0.2 Ohm to 20 Ohm
k	Boltzmann constant	J/K	$1.381 \cdot 10^{-23}$ J/K
n	Diode's ideality factor	unitless	1–2
I	PV's output current	A	$0 \le IN_p \le I_{sc}N_p$
V	PV's output voltage	V	$0 \le VN_s \le V_{oc}N_s$
P	PV's output power	W	$0 \le P_nN_pN_s$
q_e	Elementary charge	C	$1.602 \cdot 10^{-19}$ C
$\mu_{I_{sc}}, \mu_{V_{oc}}$	Temperature coefficients for current and voltage, respectively	$(I/{}^\circ C, V/{}^\circ C)$	0.04 to 0.5, -0.3 to -0.5
G, G_{ST}	Solar irradiance and irradiance at standard conditions (1000 W/m^2)	W/m^2	<1000 W/m^2, 1000 W/m^2
T, T_{ST}	Temperature of PV module and standard test values for temperature (25 °C)	°C	25 °C to 50 °C , 25 °C
$V_{oc}, V_{oc,ST}$	Voltage of PV module and standard test values for temperature (25 °C)	V	$0 \le V_{oc}N_s \le V_{oc,st}N_s$
$I_{sc}, I_{sc,ST}$	Current of PV module and standard test values for temperature (25 °C)	A	$0 \le I_{sc}N_p \le I_{sc,st}N_p$

3.2. Maximum Power Point Tracking

The output power of a PV module is given by Joule's law [24] and the I–V and P–V diagrams are presented in Figures 2 and 3:

$$P = VI, \tag{7}$$

where P, V, and I are the output power, voltage, and current of the PV module, respectively. According to (4), (6) can be rewritten as follows:

$$P = N_p I_{ph} V - N_p I_s V \left[e^{\left(\frac{q_e(V + I R_s)}{nkTN_s} \right)} - 1 \right] - \frac{V^2 + I R_s V}{R_p}, \tag{8}$$
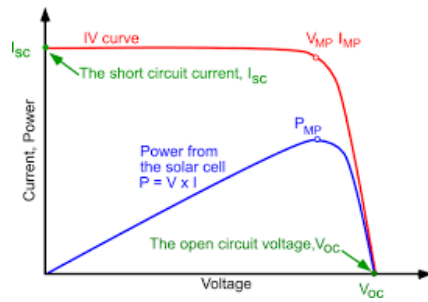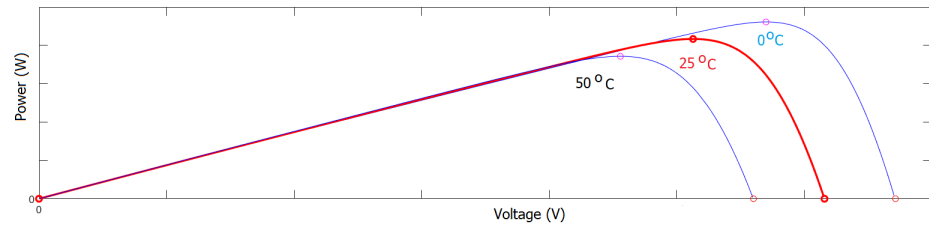
Figure 2. I–V curve.

Figure 3. P–V curve.

The PV module's output voltage and current of the photovoltaic module have a maximum value depending on irradiance and temperature. The voltage's maximum value results when no electric loads are connected to the module's output (open-circuit situation), and the current's maximum value results when the module's output is short-circuited (short-circuit situation). The equations that connect the short circuit current (output current's maximum value) and the open circuit voltage (output voltage's maximum value) with irradiance and temperature are [25].

$$I_{sc} = I_{sc}(G, T) = \frac{G}{G_{ST}} \left[I_{sc,ST} + \mu_{I_{sc}} (T - T_{ST}) \right] \tag{9}$$

$$V_{oc} = V_{oc}(G, T) = V_{oc}(T) + nV_t \ln\left(\frac{G}{G_{ST}}\right), \tag{10}$$

where

$$V_{oc}(T) = V_{oc,ST} + \mu_{V_{oc}} (T - T_{ST}), \tag{11}$$

where the following apply:

- $G_{ST} = 1000 \text{ W/m}^2$, $T_{ST} = 25 \text{ °C}$, $I_{sc,ST}$, $V_{oc,ST}$ are the standard test values for irradiance, temperature, short circuit current, and open circuit voltage, respectively.
- G, T, I_{sc}, V_{oc} are the actual values for the irradiance, temperature, short circuit current, and open circuit voltage, respectively.
- $\mu_{I_{sc}}$, $\mu_{V_{oc}}$ are the temperature coefficients for current and voltage, respectively.

The power output of the photovoltaic module is maximized, according to (4). The solution has only one global maximum point, the maximum power point (MPP), and it depends on irradiance and temperature. The electric qualities at MPP are named maximum power voltage (V_{mpp}), maximum power current (I_{mpp}), and maximum power (P_{mpp}).

Output current, output power, and, therefore, MPP are proportional to the photo-current I_{ph}, which is heavily dependent on radiation and temperature, according to the following equation:

$$I_{ph} = \frac{G}{G_{ST}} \left[I_{ph,ST} + \mu_{I_{sc}} (T - T_{ST}) \right], \tag{12}$$

where $I_{ph,ST}$ is the photocurrent under standard test conditions. The equations that combine the saturation current I_s, series resistance R_s, and the parallel resistance R_p with irradiance and temperature can be illustrated as follows [25]:

$$R_p = R_p(G) = \frac{G_{STC}}{G} \cdot R_{p,STC}, \tag{13}$$

$$R_s = R_s(G,T) = \frac{T}{T_{STC}} \left(1 - 0.217 \cdot \ln \frac{G}{G_{STC}}\right) \cdot R_{s,STC}, \tag{14}$$

$$I_s = I_s(G,T) = \frac{\left(1 + \frac{R_s}{R_p}\right) \cdot \left(\frac{G_{ST}}{G}\right) \cdot I_{sc} - \frac{V_{oc}}{R_p}}{exp\left(\frac{V_{oc}}{nV_t}\right) - exp\left(\frac{\left(\frac{G_{ST}}{G} \cdot I_{sc}\right)/R_s}{nV_t}\right)}, \tag{15}$$

where $R_{s,STC}$ and $R_{P,STC}$ are the series and parallel resistance at standard conditions, respectively, pointing out why the MPP varies in radiation and temperature.

To ensure that the PV module always produces the practicable maximum power, a converter is connected to its output. This converter controls the output voltage of the PV module to maximize the output power, according to (4) and (7). The output current can be given by (4).

If the converter is connected to the output of a combination of PV modules, then the voltage, current, and power of the PV modules are given by the following equations:

$$V_{modmpp} = \frac{V_{mpp}}{n}, \tag{16}$$

$$I_{modmpp} = \frac{I_{mpp}}{m}, \tag{17}$$

$$P_{modmpp} = \frac{P_{mpp}}{nm}, \tag{18}$$

where V_{modmpp}, I_{modmpp}, and P_{modmpp} are the output voltage, current, and power of the PV module. V_{mpp}, I_{mpp}, and P_{mpp} are the voltage, current, and power of the combination of the PV module. n and m are the number of PV modules connected in series and parallel, respectively.

3.3. Power Optimization Process

As mentioned in the previous section, all the PV parameters (e.g., I_{ph}, V_{oc}, I_{sc}, R_p, R_s) are associated with temperature and irradiance and are calculated based on PV single-diode modeling. A genetic algorithm is implemented and developed to estimate the expected (MPP) PV values according to irradiance and temperature. In essence, the genetic optimization algorithm, as detailed below, heuristically finds the MPP as well as the electric qualities at this point with regard to monitored outdoor values.

3.3.1. Fitness Function

The optimization problem consists of maximizing the power of the photovoltaic, which can be mathematically expressed as

$$\max_{I \in [0,I_{sc}],\ V \in [0,V_{oc}]} \{P\}, \text{ subject to (4)}$$

Alternatively, the fitness function can be defined as

$$\min\{-P\}, \ P = f(\mathbf{V},\mathbf{I}) = V \cdot I, f : A \to \mathbb{R}, \tag{19}$$

where input values are chosen from within the allowed set, A, as illustrated below:

$$V \in [0, V_{oc}]$$
$$I \in [0, I_{sc}]$$

and concurrently meet the appropriate constraints and requirements, namely (4), which can be also reformed to

$$I - N_p I_{ph} - N_p I_s \left[e^{\left(\frac{qe(V+IR_s)}{nkTN_s} \right)} - 1 \right] - \frac{V + IR_s}{R_p} = 0, \tag{20}$$

Exploiting the Lambert function [26,27], (4) can be modified as follows [28]:

$$I = -\frac{B}{R_s} \cdot W_o \left[\frac{A \cdot R_s}{c_1 \cdot B} \cdot e^{\left(\frac{V + R_s(A + I'_{ph})}{c_1 \cdot B} \right)} \right]$$
$$+ \frac{I'_{ph} + A - \frac{V}{R_p}}{c_1}, \tag{21}$$

where $B = n \cdot V_t$, $A = N_p \cdot I_s$, $I'_{ph} = N_p \cdot I_{ph}$, and $c_1 = 1 + \frac{R_s}{R_p}$. In that way, the PV module voltage is expressed as a function of the PV current, and constraint (20) is integrated into the fitness function. The argument of the Lambert function may take extremely large numbers due to the exponential term, leading to overflow errors in a common programming language. To overcome that problem, a technique proposed in [28] has been utilized that exploits logarithmic and exponential properties. In [28], various methods to implement the Lambert function have been compared, concluding that the "Hybrid" approach was the most accurate, providing the entire I–V curve.

According to the above modifications, the fitness function depends only on V, and (19) can be reformed as

$$\min(-P) = f(V), \tag{22}$$
$$f(\mathbf{V_{gen}}) = -V_{gen} \cdot I_{gen}(V_{gen}), \ f \ : \ [0, V_{oc}] \to \mathbb{R}$$

The parameters required for the genetic algorithm to be implemented are provided in Table 2.

Table 2. Parameters for the genetic algorithm.

Parameters	Description
p_m	mutation probability
p_c	crossover probability
population	number of chromosomes
generation	number of generations
Decimals	decimal precision of the encoded value (V_gen)
$f(\mathbf{V_{gen}}) = -V_{gen} \cdot I_{gen}(V_{gen}), \ f \ : \ [0, V_{oc}] \to \mathbb{R})$	fitness function

3.3.2. Genetic Algorithm Modeling

The genetic algorithm (GA) is a metaheuristic algorithm that was proposed by John Holland and his contributors in the 1970s [29] and is inspired by the Darwinian theory of natural selection processes [30].

Genetic algorithms are commonly used to produce high-quality solutions to search and optimization problems relying on biological-inspired operators such as selection, crossover, and mutation. The basic elements of GAs are chromosome representation, fitness selection,

and the biologically inspired operators mentioned above [31]. Usually, chromosomes are represented by a string of bits. Every chromosome contains a sequence of bits or genes, where each gene has two variant forms, 0 and 1. All chromosomes form the size of the population, and they represent different points in the solution space. The population is repeatedly replaced by the processed chromosomes using genetic operators. When all the genetic operators are applied, the population forms a new generation, and that happens constantly until the algorithm reaches the iterations that were specified by the user at an initial phase. The fitness function is a performance indicator used to evaluate how well the chromosomes are [32].

To set up the GA, all the operators are individually examined and designed depending on the solution of the problem, which is to calculate the MPPT of a PV system according to some weather conditions as well as the current and voltage that lead to the maximum power. The pseudocode for the total GA procedure is presented in Algorithm 1.

Algorithm 1 Genetic Algorithm for Minimizing $f(V) = -V \cdot I_{gen}(V)$

1: Initialize population P with random solutions
2: Evaluate the fitness of each solution in P using (22)
3: **while** termination condition not met **do**
4:     Select parents from P based on their fitness
5:     Apply crossover to create offspring
6:     Apply mutation to the offspring
7:     Evaluate the fitness of the new offspring
8:     Replace old population with a combination of parents and offspring
9: **end while**
10: **Output:** Best solution found in P

3.3.3. Encoding of Voltage

The encoding of chromosomes is the initial step in solving a problem through genetic algorithms. There are various encoding methods for GA algorithms, such as binary, permutation, value, and tree encoding. However, the encoding process is often problem-dependent, and in this work, binary encoding is used as the appropriate method. In essence, during binary encoding, each chromosome is represented by a string of bits, 0 or 1, where each chromosome bit represents a specific characteristic of the problem.

Intending to specify the length of the chromosome or the number of bits the string consists of, the domain of the objective function (i.e., the set of possible values of the independent variable) has to be defined as well as the desired decimal precision for the problem. Then, the number of bits can be calculated from the following equation [33]:

$$2^l - 1 \geq (ub - lb) * 10^{Decimals} \implies$$
$$2^l \geq (ub - lb) * 10^{Decimals} + 1 \implies$$
$$l \geq \log_2[(ub - lb) * 10^{Decimals} + 1],$$

where l is the number of bits required for the representation, *Decimals* are the decimals of precision, and *ub* and *lb* represent the upper bound and lower bound of the possible permitted values, respectively. Assuming the voltage (V) value lies in the range of $[0, 20]$ volts and one decimal of precision is required, the number of bits to represent a chromosome can be computed according to the previous equation as follows:

$$l \geq \log_2[(20 - 0) * 10^1 + 1] \implies$$
$$l \geq 7.65 \implies$$
$$l = 8$$

Taking into consideration the constraints in terms of voltage range and decimals of precision, each chromosome can be represented by a string of bits, as depicted in Figure 4. This string of bits is also called a genotype.

Figure 4. Chromosome encoding.

3.3.4. Decoding of Voltage

The decoding process is necessary in order to obtain a continuous value for the dependent variable (V), which is also required for the calculation of the objective function. The phenotype or the voltage's (V) decoded value (V_{dec}) can be retrieved using a set of equations [33].

$$precision = \frac{(ub - lb)}{2^l - 1} \tag{23}$$

The above (23) computes the precision of each bit that is used in (24) to eventually calculate the phenotype or decoded value of (V).

$$V_{dec} = \left[\sum_{n=0}^{l-1} \left(Bit \mid_{position=n} *2^n \right) \right] \cdot precision + lb \tag{24}$$

Substituting (ub, lb, l)) in (23), the precision is

$$precision = \frac{(20 - 0)}{2^8 - 1}$$
$$= 7.84 \cdot 10^{-2}$$

For the chromosome in Figure 4, the phenotype can be calculated according to (24), as follows

$$V_{gen} = V_{dec} = \left(2^3 + 2^5 \right) \cdot precision + lb$$
$$= \left(40 \right) \cdot 7.84 \cdot 10^{-2} + 0$$
$$= 3.136$$

3.3.5. Selection

During every generation, some chromosomes of the existing population are selected to later breed a new population through the crossover procedure. Chromosomes compete with each other through a fitness-based process, where the ones that better fit the fitness function are more likely to be selected to proceed. It should be noted that every chromosome can be selected more than once. Among the most popular methods that measure the fitness of the solutions to select the chromosomes is tournament selection [34], which was used in this paper.

During tournament selection, an individual from a population of chromosomes is selected. Specifically, during this process, a number of chromosomes that are randomly picked from the total population are evaluated according to their fitness, and the best solution is selected to perform crossover. This process repeats as many times as the population size. The tournament size or the participants' pool size indicates the tolerance to weak individuals since a smaller number of participants means that weak chromosomes experience a higher chance of being selected as winners of the tournament.

3.3.6. Crossover

In order to generate new offspring, the genetic information of two parents (chromosomes) is combined using the crossover genetic operator. This is the most common way to stochastically generate new chromosomes from the existing population. There are various methods for performing crossover, such as the two-point crossover that was used in this paper [35].

During the two-point crossover method, two different random integers between [0, l] are generated to specify the two crossover points for the parent chromosomes. The bits that lie between these points are swapped between the parent chromosomes, resulting in two child chromosomes, each of which carries a portion of genetic information from both parents (Figure 5).
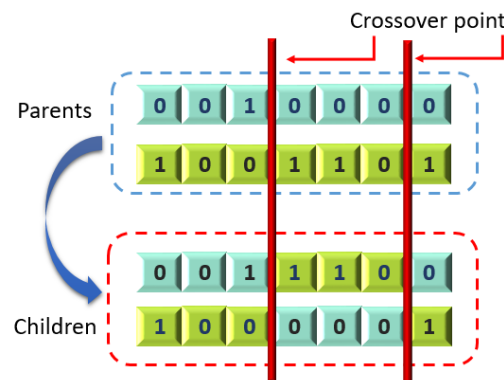

Figure 5. Two-point crossover example.

However, not all parents' chromosomes undertake the crossover process. After selection has occurred, two chromosomes (parents) are selected to perform the crossover. A given crossover probability $p_c \in [0, 1]$ is defined at the very beginning and specifies whether the crossover will be carried out. The crossover will be performed under the condition that a random number generator providing numbers between [0, 1] will generate a number smaller than p_c. All chromosomes are selected successively in a set of two until the entire population is selected.

3.3.7. Mutation

In genetic algorithms, mutation is a unary genetic operator, where, in contrast to the crossover operator, it is applied only to one chromosome and is used in order to preserve the genetic diversity of chromosomes in a population from one generation to the next.

Mutation modifies the chromosome by altering one or more genes in a chromosome. These modifications can lead to a new generation with a population of chromosomes with better solutions than the previous ones. The mutation is carried out as an evolution process according to the probability of mutation $p_m \in [0, 1]$ defined during a trial phase.

There are various methods for performing the mutation operator, the most common of which is called the flip bit [35] method, which involves a probability given by a random number generator that a bit in a chromosome will be flipped from its original state. This occurs for every bit in the sequence of the chromosome. If the probability is smaller than p_m, then the particular bit that is examined will be flipped. In Figure 6, an example of a flip mutation is depicted where only the fifth bit of the chromosome is flipped.
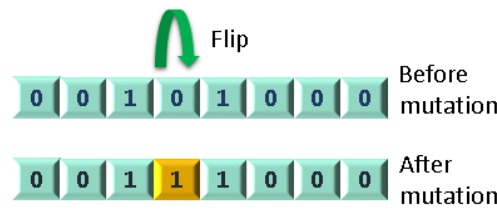
Figure 6. Flip mutation example.

3.4. PV Error Handling

The comparison between the voltage and current predicted by the genetic optimization algorithm (V_{gen}, I_{gen}) in Section 3.3 and the actual voltage and current (V_{act}, I_{act}) involves a meticulous process, as depicted in the following flow chart (see Figure 7). This flow chart illustrates the systematic evaluation of the PV module's or PV array's operation by quantifying the error between predicted and actual values. The process begins with the extraction of predicted values (V_{gen}, I_{gen}) and actual values (V_{act}, I_{act}) from the genetic optimization algorithm and experimental measurements, respectively.
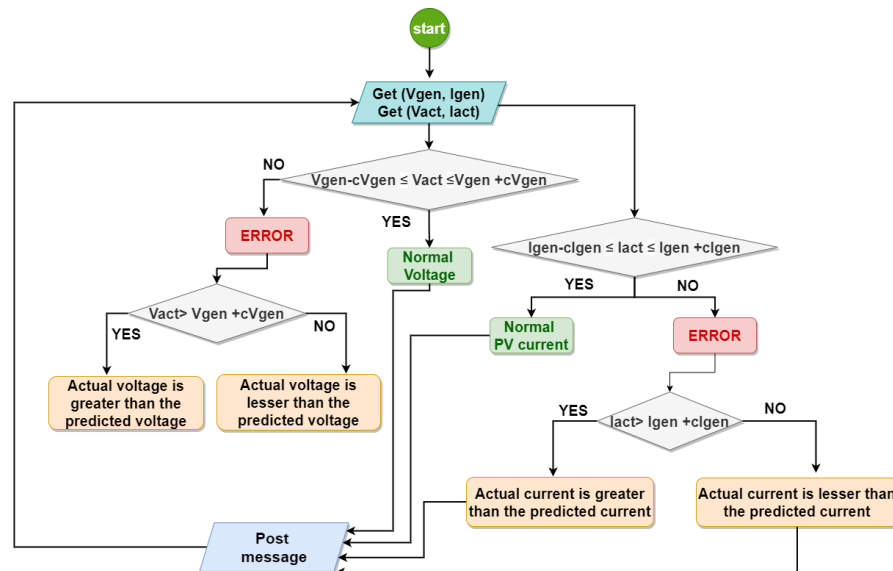

Figure 7. Flow chart of PV error handling.

The next step involves the subtraction of actual values from predicted values, resulting in the computation of the error for both voltage and current. This error is then compared against a predefined constant *c*, typically set experimentally to 0.1 for enhanced precision. The decision block assesses whether the error exceeds the established threshold. If the error surpasses the threshold, the system flags the occurrence of an anomaly or deviation from expected performance. In case the error is within the acceptable range, the system proceeds to the output phase, indicating that the PV module or array is operating within the desired parameters. This systematic comparison and error analysis serve as a critical step in validating the effectiveness and accuracy of the genetic optimization algorithm in predicting the electrical characteristics of the photovoltaic system.

The overall process encapsulated in this flow chart ensures a comprehensive assessment of the algorithm's performance, enabling researchers and engineers to identify and address discrepancies between predicted and actual values, thereby refining the modeling and optimization of the photovoltaic system (Figure 7).

This flow chart provides a visual representation of the systematic comparison and error analysis between predicted and actual values in the context of photovoltaic system modeling and optimization. The pseudocode for the PV error-handling algorithm is presented below (Algorithm 2).

Algorithm 2 PV Error Handling

1: **function** HANDLEPVERROR(V_{gen}, I_{gen}, V_{act}, I_{act}, threshold)
2:   ▷ Calculate error for voltage and current
3:     $error\_V \leftarrow V_{gen} - V_{act}$
4:     $error\_I \leftarrow I_{gen} - I_{act}$
5:   ▷ Check if error exceeds the predefined threshold
6:     **if** $|error\_V| >$ threshold or $|error\_I| >$ threshold **then**
7:   ▷ Anomaly detected, flag the occurrence
8:         $system\_status \leftarrow$ "Anomaly Detected"
9:     **else**
10:   ▷ System operates within desired parameters
11:        $system\_status \leftarrow$ "Normal Operation"
12:    **end if**
13:   ▷ Output the results
14:    **return** $system\_status, error\_V, error\_I$
15: **end function**

3.5. Overall Architecture

The overall methodology, as depicted in Figure 8, details the process of comparing and validating actual values with predicted values in the context of PV system modeling and optimization. The essential parameters guiding this model are thoughtfully cataloged in Table 3, offering a comprehensive overview of the variables involved in the comparison process. All crucial parameters, including I_{gen}, V_{gen}, P_{gen}, I_{act}, V_{act}, and P_{act}, play a distinct role in the validation procedure. I_{gen} and V_{gen} represent the output current and voltage, respectively, calculated from the genetic algorithm, providing predicted values for the installed PV system. Similarly, P_{gen} signifies the output power generated by the PV system, as determined by the genetic algorithm. On the other side, I_{act}, V_{act}, and P_{act} denote the measured values of output current, voltage, and power from the PV system meters, offering a real-world reference for the system's performance.
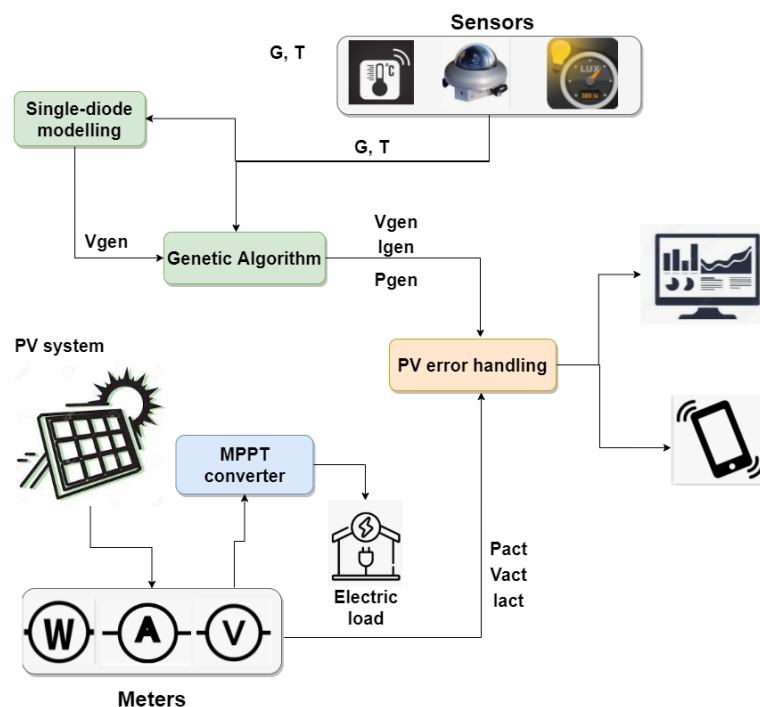

Figure 8. PV error-handling conceptual architecture.

Table 3. Parameters to compare actual values from predicted.

Parameters	Descritpion	Unit
I_{gen}	Output current of installed PVs as calculated from genetic algorithm	A
V_{gen}	Output voltage of installed PVs as calculated from genetic algorithm	V
P_{gen}	Output power of installed PVs as calculated from genetic algorithm	W
I_{act}	Output current of installed PVs as measured from meters	A
V_{act}	Output voltage of installed PVs as measured from meters	V
P_{act}	Output power of installed PVs as measured from meters	W

The comprehensive procedure unfolds as follows: leveraging data from sensors and meters and exploiting the single-diode model, the GA produces the predicted values (V_{gen}, I_{gen}, P_{gen}). Simultaneously, the actual output voltage, current, and power (V_{act}, I_{act}, P_{act}) are retrieved from the meters embedded within the PV system. These actual and predicted values then undergo a comparison through the PV error-handling algorithm, as explicated in the preceding section. This algorithm systematically assesses the deviation between actual and predicted values, considering predefined thresholds for error, typically set experimentally to enhance accuracy. The outcome of this analysis provides invaluable insights into the efficacy of the genetic algorithm in modeling and optimizing the PV system, allowing researchers and engineers to refine and improve the accuracy of their models.

The overall procedure is described in Algorithm 3.

Algorithm 3 Real-time PV Error Handling

Require: I_{gen}, V_{gen}, P_{gen}, I_{act}, V_{act}, P_{act}
 1: **function** REALTIMEERRORHANDLING(I_{gen}, V_{gen}, P_{gen}, I_{act}, V_{act}, P_{act})
 2:     **for** each parameter *param* in $\{I, V, P\}$ **do**
 3:         *error* $\leftarrow$ abs($param_{gen} - param_{act}$)   ▷ Calculate absolute error
 4:         **if** *error* exceeds predefined threshold **then**
 5:             **Alert:** High deviation in *param* values
 6:         **end if**
 7:     **end for**
 8:     **Return:** Analysis results and insights
 9: **end function**

---

4. Results

In this section, the results of the proposed method are presented. Initially, the performance and accuracy of the proposed GA applied to model single-diode PV systems under various illuminance conditions are addressed. The PV parameters used in the experiments are outlined, and the sensors and meters used are presented while detailing the overall procedure that involves measuring real PV values, expected MPP values through the GA, and the subsequent comparison. The GA's accuracy is investigated by altering parameters affecting PV output, such as temperature and irradiance. Additionally, the GA's key hyperparameters, including population size, number of generations, crossover and mutation probabilities, are thoroughly tested to optimize its performance. The accuracy and efficiency of the GA are further compared with other optimization algorithms, showcasing the GA's efficacy in solving complex PV optimization problems. The results highlight the GA's ability to provide accurate predictions and its suitability for real-time PV monitoring and optimization, making it a valuable tool for practical applications.

4.1. Experimental Set Up

In this experimental set-up, diverse illuminance conditions were applied to assess the performance of PV systems, with the parameters of the utilized PVs detailed in Table 4. The table provides a comprehensive overview of critical PV parameters essential for accurate modeling and optimization. The experiment's focus is on understanding how the PVs respond to varying illuminance levels, shedding light on their efficiency and robustness under different environmental scenarios. By systematically varying illuminance conditions, the study aims to capture the nuanced behavior of the PV systems and evaluate their real-world adaptability. The parameters outlined in Table 4, such as the diode ideality factor (n), series resistance (R_s), and photogenerated current ($I_{ph,ST}$), play pivotal roles in determining the system's response to changing illuminance. The ensuing analysis and results will provide valuable insights into the dynamic behavior of PV systems, contributing to a broader understanding of their performance in varying environmental contexts.

Table 4. Parameters of the PVs used in the experiments.

PV Parameter	Value	Unit
n	0.988	unitless
R_p	249.678	*Ohm*
R_s	0.384	*Ohm*
$I_{sc,ST}$	5.17	A
$I_{ph,ST}$	5.178	A
$V_{oc,ST}$	43.99	V
$\mu_{I_{sc}}$	0.00415	$I/^\circ C$
$\mu_{V_{oc}}$	-0.03616	$V/^\circ C$
N_s	72	unitless
I_s	1.784×10^{-10}	A

Sensors and Meters

The comprehensive modeling procedure involves acquiring crucial measurements from an array of devices, meters, and sensors to accurately assess the performance of PV systems. The key objective is to identify potential errors in PV operations by comparing real PV values with the expected MPP values. Real PV values are determined through the utilization of meters and MPPT, while the expected PV values are estimated using a genetic algorithm.

For an accurate simulation of PV system operation, the GA relies on input values from various outdoor sensors and PV modeling. These inputs include temperature measurements obtained through type J thermocouples [36] attached to the PV collectors, providing insights into temperature variations during operation. Additionally, an optical sensor [37] is employed to simultaneously monitor the brightness of the surrounding area, contributing to a comprehensive understanding of the environmental conditions affecting PV performance. A pyrometer sensor is incorporated to measure solar radiation, offering reliable data on solar energy incidence.

Simultaneously, the monitoring system utilizes a multimeter to track the voltage and current values of the PV collectors, providing real-time data on electrical parameters. Simultaneously, a wattmeter is employed to monitor the power produced by the solar cells, offering insights into the overall energy generation of the PV system. To convert analog signals into digital values for precise measurements, an analog-to-digital converter (ADC) is employed, facilitating the retrieval of accurate and digitized data. The final digital outputs include the actual electric qualities, namely V_{act} (voltage), I_{act} (current), and power P_{act}, forming the basis for the subsequent analysis and comparison with the expected values generated by the genetic algorithm. This meticulous data acquisition process ensures a comprehensive and accurate assessment of the PV system's real-time performance and aids in the identification of any deviations or errors in its operation.

4.2. Genetic Algorithm Tests

In this section, the focus is on the comprehensive exploration of GA tests and their impact on the modeling accuracy of PV systems under various conditions. The population size and number of generations, pivotal parameters in GAs, are investigated to optimize convergence and computational efficiency. Results from performance tests will present the influence of these parameters on the GA's execution time and convergence ability. Additionally, the critical selection of crossover and mutation probabilities is addressed through experimental tests, revealing their relationship and impact on the algorithm's performance. This analysis will lead to the identification of an optimal combination of hyperparameters for the GA crossover probability. Subsequently, the accuracy of the GA is thoroughly examined through tests where irradiance and temperature variations are introduced.

4.2.1. Population Size and Number of Generations

The selection of the population size is a crucial process for the GA because it directly influences the convergence of the algorithm [38]. As a result, population size has an impact on the GA's ability to reach the optimal solution [39]. According to the findings in the literature, a small population mars the quality of chromosomes, and a large population size (i.e., more than 100) is only a computational waste [40]. Nonetheless, to determine the appropriate population and the number of generations (iteration times), tests were carried out, as shown in Table 5. The crossover probability was set to 0.75, and the mutation probability was set to 0.01, as suggested in the literature [41]. It could be suggested that the experiments indicate that a population size of 60 is efficient for the GA. Furthermore, a small generation number will maintain the quality of the solutions and keep the execution time rather low.

Table 5. GA performance tests on the population and the generation size.

Generation	Population	Execution Time (s)	Convergence
	10	0.052	8/10
10	20	0.110	10/10
	40	0.229	10/10
	10	0.112	9/10
20	20	0.229	10/10
	40	0.477	10/10
	10	0.237	10/10
40	20	0.484	10/10
	40	0.971	10/10
	10	0.479	10/10
80	20	0.951	10/10
	40	1.928	10/10

Some experiments have been conducted to measure the computational difficulty of the algorithm and to associate some of the genetic hyperparameters (population size and generations) with the execution time while keeping crossover and mutation probabilities intact. According to Table 5, and observing how the execution time varies as the population and generations increase, it can be observed that the execution time is linearly proportional to the size of the population. Moreover, the slope of the straight lines in Figure 9 indicates that the execution time is also linearly proportional to the number of generations. Combining the impact of those two hyperparameters on the execution time and considering

that the genetic algorithm is heuristic and crossover/mutation operators occur at nearly the same time, the algorithm has a complexity of $O(n^2)$.
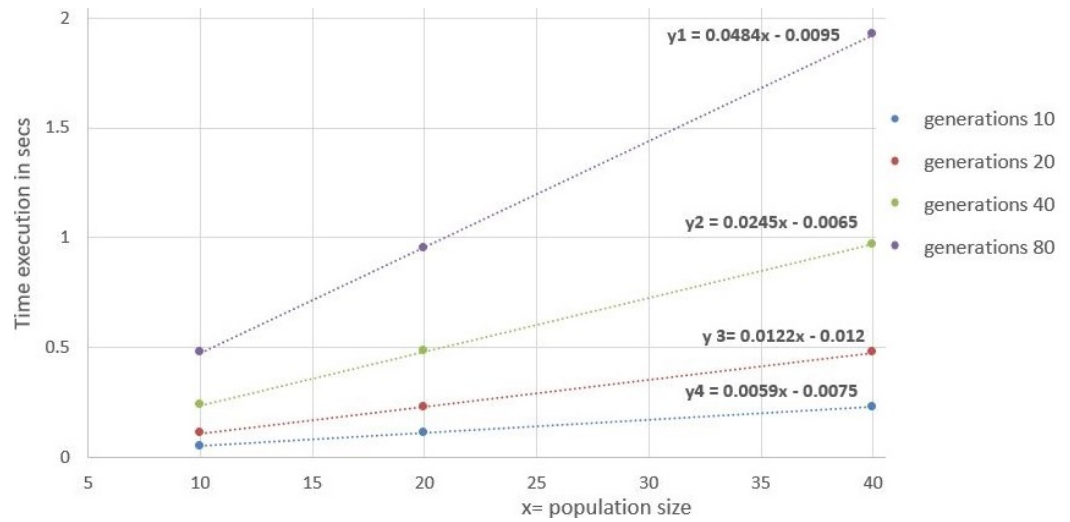

Figure 9. Relationship between population size, generations, and execution time.

As depicted in Figure 9, there is an observable relationship between the population size and the execution time of the GA. Furthermore, there is also a proportional relationship between execution time, generation size, and population size. Regarding the tests developed, the aforementioned relationships are estimated by the following:

$$y_1 = 0.0484x - 0.0095$$
$$y_2 = 0.0245x - 0.0065$$
$$y_3 = 0.0122x - 0.012$$
$$y_4 = 0.0059x - 0.0075,$$

where x is the population size and y is the execution time in seconds, where y_1, y_2, y_3, y_4 correspond to generation sizes 10, 20, 30, 40, respectively.

4.2.2. Crossover and Mutation Probability

The crossover probability and the mutation probability are also very important parameters for the GA [30]. They should be selected after a trial phase where various combinations must be tested. The proposed GA was tested for many combinations. The results are presented in Figure 10, the two red lines indicate the lowest value of the objective function and the value of the fitness function when the algorithm stops. When there is one line, the GA converges to the minimal point.
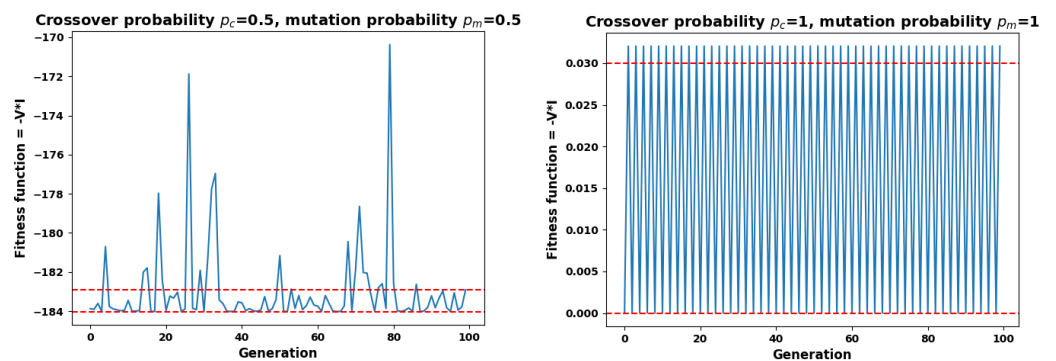

Figure 10. Experimental tests for GA performance setting crossover probability and mutation equal to 0.5 and 1.0.

To start with, two rather rare combinations were selected where crossover probability and mutation were equal and set to 0.5 and 1.0. In the first case, it is observed that the GA does not converge until generation 100, and the algorithm often has high picks. This is due to the high rate of both selected probabilities. In the second case, where the probabilities are set to 1.0, it may be observed that the algorithm seems to be mixed up and going all over the search space.

Subsequently, to test the effect of the crossover probability on the GA, the mutation probability is kept fixed and equal to $p_m = 0.8$. As it may be observed in Figure 11, the GA never converges with a high mutation probability, regardless of the p_c value. The GA had sharp fluctuations during the 100-generation deployment. It is suggested that a high rate of mutation disorganizes the GA.
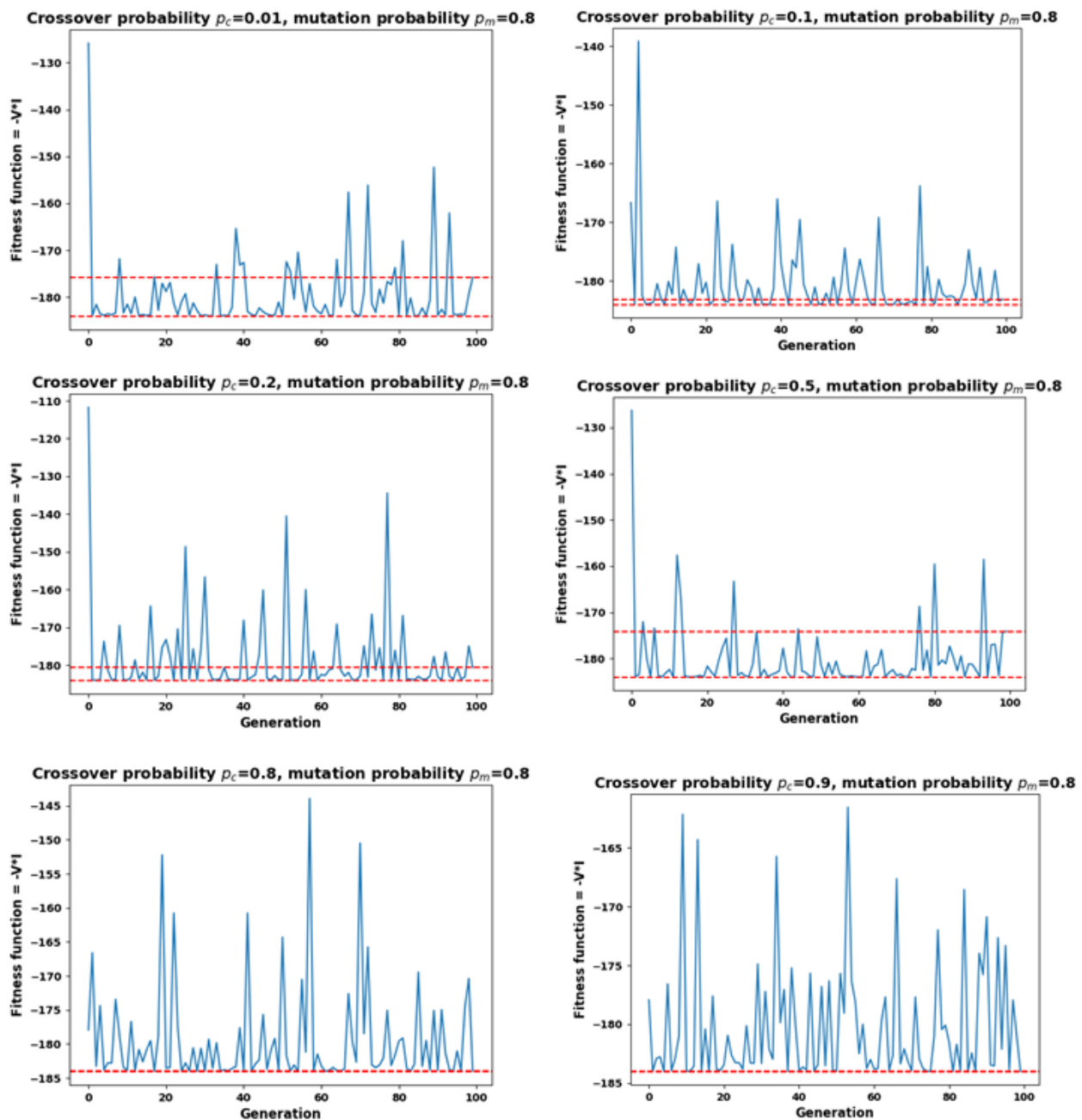
Figure 11. Crossover and mutation tests with mutation fixed to $p_m = 0.8$.

Moreover, to test the effect of the mutation probability on the GA, the crossover is kept fixed and equal to $p_c = 0.8$. In contrast to the previous case where the p_m was high, when the p_c is high, the GA generally performs better (Figure 12). Furthermore, the lower the p_m, the better. A p_m greater than 0.05 starts to deteriorate the GA's ability to converge to the optimal solution.
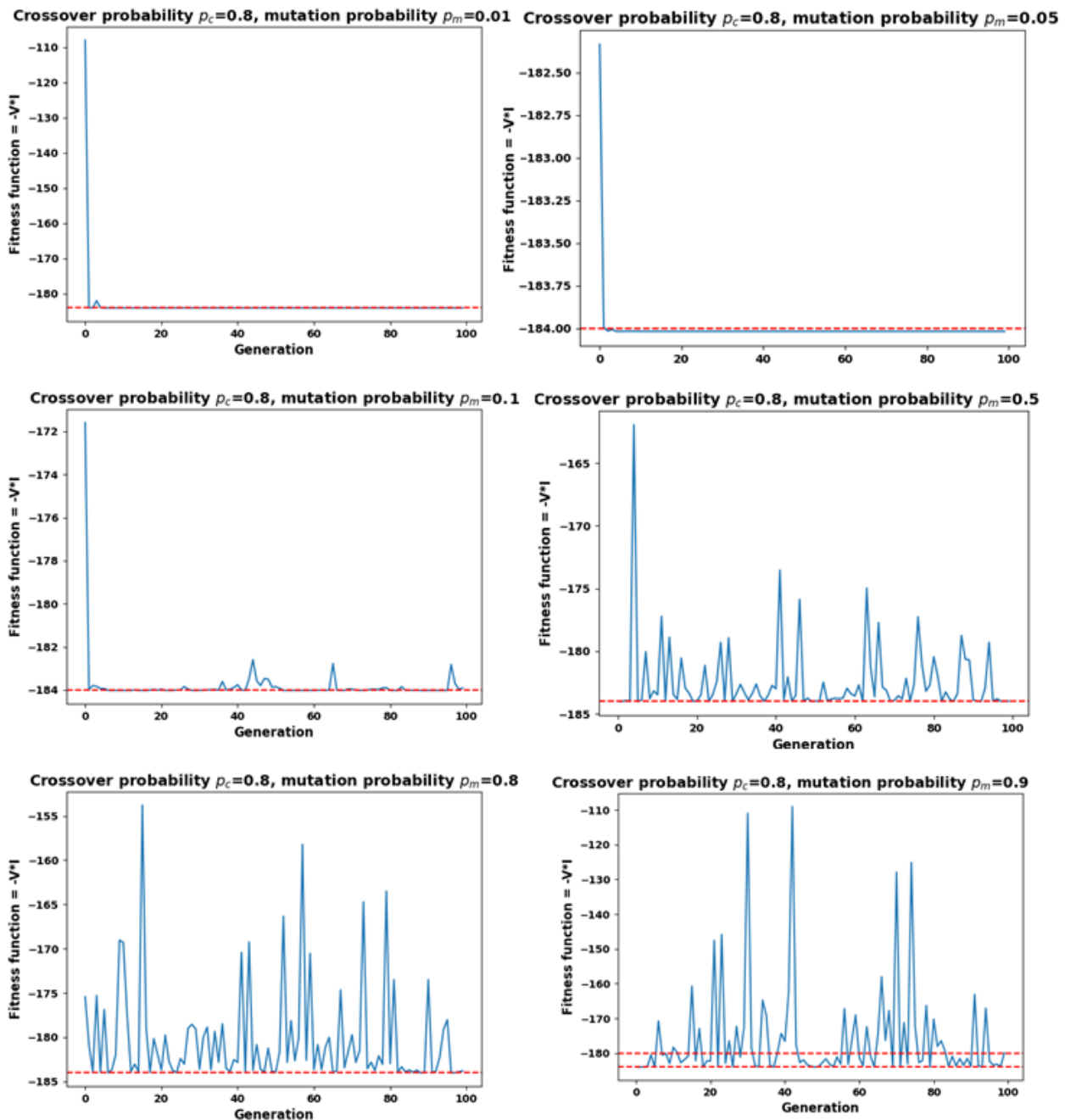

Figure 12. Crossover and mutation tests with crossover fixed to $p_c = 0.8$.

To sum up, the best performance of the suggested GA is observed when $p_c = 0.8$ and $p_m = 0.01$. Therefore, this is the combination that is used for the GA because it ensures that the GA will find the best solution while quickly converging.

4.3. GA Accuracy

After specifying the crossover and mutation probability and setting the decimal precision to two, some experiments have been performed to measure the accuracy of the

genetic algorithm. Specifically, by modifying the parameters that mostly contribute to the PV output (temperature and irradiance), the genetic algorithm was executed and the predicted values of the electric quantities were compared to the real ones, as measured by the meters. Holding irradiance at a constant value and modifying temperature, three tables of results were formed for three different values of irradiance, namely (214.72, 500, 1000). The results can be easily depicted in Tables 6–9. In most cases, the genetic algorithm is more than 98% accurate.

Table 6. GA accuracy tests with irradiance 214.72 W/m^2.

Irradiance 214.72 W/m^2			
Temperature 0 °C			
	Real Values	**Genetic**	**Error**	**Accuracy %**
I_{mpp} **A**	0.990	1.01223	−0.02223	98
V_{mpp} **V**	33.540	34.5158	−0.97582	97
P_{mpp} **W**	33..2	34.93794	−1.73794	95
Temperature 25 °C			
I_{mpp} **A**	1.000	1.02635	−0.02635	97
V_{mpp} **V**	29.93.	32.42978	−2.49978	92
P_{mpp} **W**	29.930	33.28430	−3.35435	89
Temperature 50 °C			
I_{mpp} **A**	1.025	1.03863	−0.01363	99
V_{mpp} **V**	25.810	30.37843	−4.56843	82
P_{mpp} **W**	26.450	31.5519	−5.10194	81

Table 7. GA accuracy tests with irradiance 500 W/m^2.

Irradiance 500 W/m^2			
Temperature 0 °C			
	Real Values	**Genetic**	**Error**	**Accuracy %**
I_{mpp} **A**	2.44	2.36011	0.07989	97
V_{mpp} **V**	38.58	37.22944	1.35056	96
P_{mpp} **W**	94.13	87.86557	6.26442	93
Temperature 25 °C			
I_{mpp} **A**	2.41	2.39144	0.01856	99
V_{mpp} **V**	35.96	34.8984	1.0616	97
P_{mpp} **W**	86.81	83.45743	3.35257	96
Temperature 50 °C			
I_{mpp} **A**	2.46	2.41856	0.04144	98
V_{mpp} **V**	31.13	32.58600	−1.456	95
P_{mpp} **W**	76.57	78.81119	−2.24119	97

Table 8. GA accuracy tests with irradiance 1000 W/m^2.

	Irradiance 1000 W/m^2		
	Temperature 0 °C		
	Real Values	**Genetic**	**Error**	**Accuracy %**
I_{mpp} **A**	4.800	4.74547	0.05453	99
V_{mpp} **V**	40.110	39.28088	0.82912	98
P_{mpp} **W**	192.550	186.40623	6.14376	97
	Temperature 25 °C		
I_{mpp} **A**	4.830	4.77966	0.05034	99
V_{mpp} **V**	36.220	36.63148	−0.41148	99
P_{mpp} **W**	174.984	175.08602	−0.10202	100
	Temperature 50 °C		
I_{mpp} **A**	4.790	4.82245	−0.03245	99
V_{mpp} **V**	32.530	33.81646	−1.28646	96
P_{mpp} **W**	155.820	163.07818	−7.25818	95

Table 9. Output results of SLSQP, COBYLA and Rao (1,2,3) algorithms.

	Test Results					
	Irradiance 1000 W/m2					
	Temperature 0 °C					
	Real Values	**GA**	**SLSQP**	**Cobyla**	**Rao-1**	**Rao-2**	**Rao-3**
Impp	4.80	4.75	4.72	3.82	4.74	4.73	4.73
Vmpp	40.11	39.28	39.49	43.65	39.45	39.48	39.48
Pmpp	192.55	186.41	186.48	166.79	187.20	186.50	186.50
	Temperature 25 °C					
Impp	4.83	4.78	4.78	3.30	4.80	4.79	4.79
Vmpp	36.22	36.63	36.63	42.12	36.59	36.62	36.62
Pmpp	174.98	175.09	175.10	138.78	175.02	175.08	175.08
	Temperature 50 °C					
Impp	4.79	4.82	4.82	2.87	4.81	4.82	4.82
Vmpp	32.53	33.82	33.82	40.68	33.78	33.81	33.81
Pmpp	155.82	163.08	163.10	116.88	162.95	163.09	163.09

It is worth mentioning that, in Table 6, for a low irradiance value (214.72), it can be easily observed that, as temperature increases, the accuracy of the algorithms decreases. In particular, when the temperature equates 50 °C, the real value diverges from the predicted one by 18%. That noticeable deviation between the real and predicted values is not attributed to the genetic algorithm but to the PV modeling. PV parameters depend on temperature and irradiance but are usually not taken into account in the equations to simplify the modeling process, leading to less accurate solutions when irradiance or temperature diverge a lot from the standard conditions. Various PV modeling methods are proposed in [25].

On the contrary, in Table 7, when the irradiance value (500 W/m^2) is closer to that of the standard conditions, better results were obtained, while the precision in every quantity

of electrics does not fall below 96%. Similar results can be illustrated in Table 7, where accuracy remains above 95%. Due to the fact that electric quantities (voltage and current) can be predicted with an error less than 10%, except for cases of extreme values of irradiance and temperature, an error could be raised when real and predicted values diverge more than 10%. It should also be remembered that, with more analytical and detailed PV modeling, the genetic algorithm could yield better results, even in extreme cases.

4.4. PV Error Generation

The end-user can be informed of the PV's state from the real-time analytics platform implemented to provide valuable information about the PVs. In particular, beyond other crucial and informative visualizations, the end-user also has the ability to constantly view the real-time ground truth and predicted values of the electric quantities and check if any abnormal activity is observable. Two different cases of normal and abnormal PV activity can be illustrated in Figures 13 and 14, respectively. It should also be mentioned that, in cases of abnormal activity, pop-up notifications also appear on the user's screen.
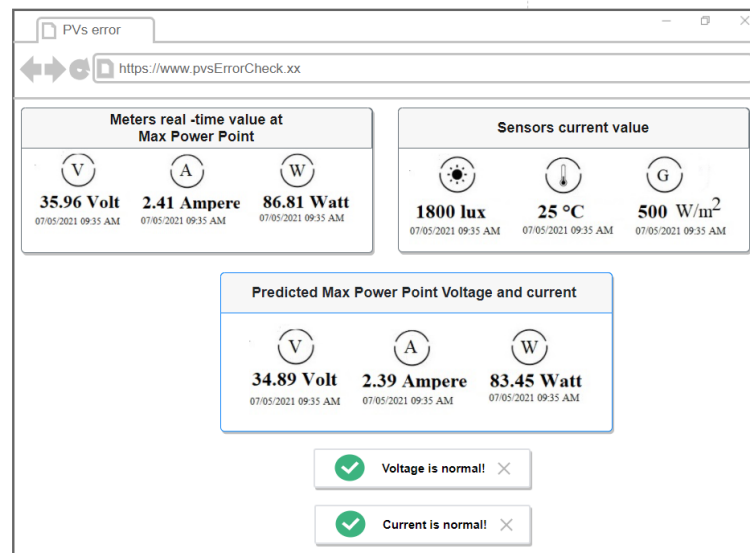

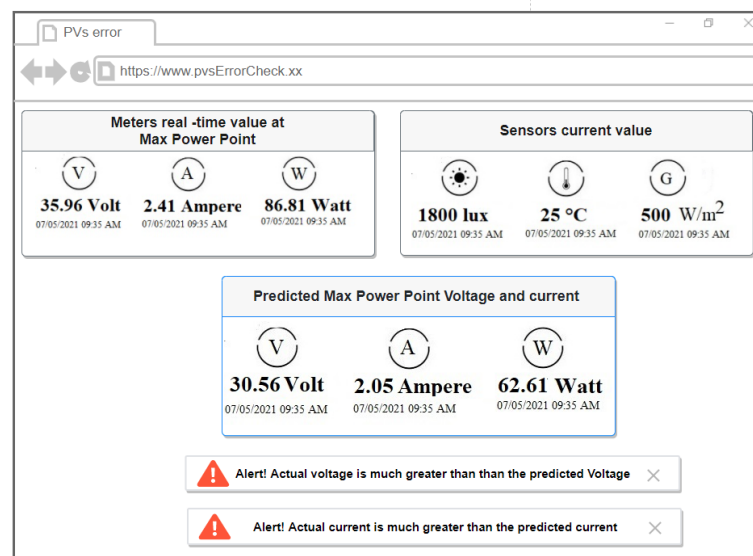**Figure 13.** UI example when PV values are normal.

Figure 14. UI example when PV values are not normal.

4.5. Comparing Proposed Fitness Function with MPC-Based Methods

It has been proposed that MPC-based MPPT algorithms are of high accuracy and robustness [30]. With this technique, the MPPT is tracked by correcting V and I values until $\frac{dP}{dV} = 0$. Exploiting the Lambert function and finding I(V), as in (21), the above equation can be reformed and is illustrated as

$$\frac{dP}{dV} = \frac{d(V \cdot I)}{dV} = I(V) + V \cdot \frac{dI(V)}{dV} = 0 \tag{25}$$

Using the above (25), a fitness function to find the MPP was formed and is expressed as follows:

$$f(\mathbf{V_{gen}}) = |I(V) + V \cdot \frac{dI(V)}{dV}|,\ f\ :\ [0, V_{oc}] \to \mathbb{R}$$

The genetic algorithm was also tested with the above fitness function, and there were no substantial deviations between the results using the fitness function defined in (22). However, $\frac{dI(V)}{dV}$ should be calculated, making the above fitness function much more complex than the proposed solution in this paper. Speaking of complexity, $\frac{dI(V)}{dV}$ can be calculated using the Lambert function and (21), as follows:

$$\frac{dI(V)}{dV} = -\frac{V_t}{R_s} \cdot W_o \left[\frac{(I_s \cdot R_s)}{(1+\frac{R_s}{R_{sh}}) \cdot V_t} \cdot e^{\frac{V+R_s(I_s+I_{ph})}{(1+\frac{R_s}{R_{sh}}) \cdot V_t}} \right]$$
$$+ \frac{(I_{ph}+I_s-\frac{V}{R_{sh}})}{1+\frac{R_s}{R_{sh}}} \tag{26}$$

To summarize, the solution proposed in this paper is not only less complex than other solutions to track the MPP, but the genetic algorithm also offers the flexibility to use another technique (e.g., MPC, as above) by modifying the fitness function.

4.6. Genetic Algorithm Performance Test Compared with Other Optimization Algorithms

There are a multitude of numerical optimization algorithms in the literature that perform in various ways [42]. Their performance varies depending on their formulation (e.g., convergence velocity, optimum perception, constraint handling). There are three main categories of optimization algorithms: evolutionary algorithms, gradient-based algorithms, and non-gradient-based algorithms [43]. During the design process of a problem, the appropriate algorithm must be selected to formulate a plausible solution. The meticulous choice of the optimization algorithm is critical for the robustness of every problem-solving process. As a result, the created genetic algorithm (i.e., evolutionary) is compared to two other algorithms belonging to the other categories and one belonging to the same category.

The first one is a gradient-based algorithm, sequential least squares programming (SLSQP) [44]. SLSQP exploits the Han–Powell quasi-Newton method with an update of the B-matrix [44]. It is a commonly used algorithm that handles problem constraints with efficacy. The other one is a non-gradient-based algorithm, COBYLA [45]. COBYLA does not account for bounds and is a non-derivative optimizer [45]. Therefore, the created GA will be examined both with a gradient and with a non-gradient algorithm using the boundary constraints of the PV configuration. As far as the evolutionary algorithm is concerned, in 2020, Ravipudi Venkata Rao proposed the Rao algorithms (i.e., Rao-1, Rao-2, Rao-3), which are metaphor-free optimization methods using best/worst solutions and random interactions for quick convergence [46]. They require only two parameters—population size and maximum evaluations—streamlining the tuning process by omitting complex parameters from previous metaphor-based approaches. All the algorithms are tested for their performance compared to real PV values and the same input parameters (i.e., irradiance, temperature). Moreover, two ways of handling constraints are compared: the one used by GA exploiting the Lambert function, and the other used by SLSQP, COBYLA, and Rao (1, 2, 3), which uses the domain of (4).

The comparison results in Tables 9–11 reveal that the GA, SLSQP, and COBYLA algorithms yield nearly identical results across different temperatures and output parameters. This uniformity underscores the accuracy of the constraint handling, particularly the use of the Lambert function in the GA. Despite the algorithms' similar performance, the GA stands out as a preferable choice due to its configurability. The GA offers more flexibility in selecting various objective functions and constraints, providing adaptability to solve a unique problem. This versatility in configuration is a significant advantage, allowing the GA to address a wide range of optimization scenarios effectively. As for the time required for each algorithm to achieve convergence, the results indicate that the GA consistently demonstrates a slightly more efficient convergence across varying temperatures (0 °C, 25 °C, and 50 °C) compared to the SLSQP, Cobyla, and Rao-1, Rao-2, and Rao-3 algorithms. The advantage of the GA in convergence time suggests its robust performance in optimizing the given parameters, highlighting its potential as an effective optimization method for the specified conditions.

Table 10. Execution time comparison of optimization algorithms (in milliseconds).

Temperature (°C)	GA	SLSQP	Cobyla	Rao-1	Rao-2	Rao-3
0	100	105	108	120	106	107
25	90	95	95	115	96	97
50	98	105	105	105	104	105

Table 11. Comparison results of SLSQP, COBYLA, and Rao (1,2,3), based on the outputs.

	GA %	SLSQP %	Cobyla %	Rao-1 %	Rao-2 %	Rao-3 %
Temperature 0 °C						
Impp	99	98	80	99	99	99
Vmpp	98	98	92	98	98	98
Pmpp	97	96	86	96	96	96
Temperature 25 °C						
Impp	99	99	69	98	98	98
Vmpp	99	99	84	99	99	99
Pmpp	100	100	80	100	100	100
Temperature 50 °C						
Impp	99	99	60	99	99	99
Vmpp	96	96	75	96	96	96
Pmpp	95	95	75	96	96	96

Furthermore, the GA's ease of application becomes evident in comparison to model predictive control techniques [47]. The GA's straightforward implementation and ability to handle diverse problem formulations make it a practical and accessible choice for optimization tasks in this context. In contrast to the Rao algorithms, the GA consistently demonstrates slightly better performance across all tested conditions. This superiority, combined with the GA's configurability and ease of use, positions it as the method of choice for optimizing the given system.

4.7. Results and Discussion

The proposed methodology combines advanced computational techniques, experimental measurements, and error-handling strategies to model and optimize photovoltaic (PV) systems. The integration of a GA with a single-diode model forms the basis of the

modeling approach. The ensuing discussion delineates the key aspects and implications of this methodology.

The utilization of a GA involves a meticulous examination of its performance and sensitivity to various parameters. The GA's efficacy is contingent upon optimal hyperparameter tuning, particularly concerning population size, generation number, crossover probability, and mutation probability. The comprehensive error handling and validation procedure establishes a robust mechanism for comparing predicted and actual PV system outputs. The integration of a predefined error constant facilitates a quantitative assessment of the algorithm's performance. The discussion on the error-handling algorithm emphasizes its utility in identifying discrepancies and guiding further refinement of the genetic algorithm.

The integrated procedure, harmonizing GA predictions with actual measurements, offers a holistic perspective on PV system modeling. The step-by-step approach, encompassing sensor data, GA predictions, and metered values, ensures a comprehensive evaluation. The methodology's practical implications lie in its ability to not only optimize PV system performance but also provide a systematic framework for continuous improvement.

While the proposed methodology exhibits promising results, it is not without limitations. The simplified single-diode model may introduce inaccuracies, especially in extreme conditions. Future research could explore more intricate PV models to enhance accuracy. Additionally, the sensitivity of the GA to initial conditions warrants further investigation for robust real-world applicability.

5. Conclusions

In this paper, a novel evolutionary algorithm has been presented for real-time PV power optimization, introducing a comprehensive system that monitors and maintains the health of PV installations. The system provides a holistic overview of outdoor environmental conditions, continuously monitors PV voltage, power, and current, and predicts PV values in real time. Alerts and notifications are sent to stakeholders and end-users, enhancing the practical utility of the system.

The power optimization process underwent a thorough analysis, leading to the development of a finely tuned GA. Through a systematic trial-and-error phase, optimal GA parameters were determined—$p_c = 0.8$, $p_m = 0.01$, and a population size of 60—resulting in a remarkable accuracy exceeding 98% when compared to actual PV values. This underscores the effectiveness of the proposed GA in optimizing PV power output.

To ensure the operational integrity of PV modules or arrays, a robust comparison between actual and predicted voltage and current was implemented. The proposed error-handling procedure demonstrated efficiency during experimentation, promptly alerting end-users to potential PV malfunctions. A comparative analysis with other algorithms affirmed the accuracy of the genetic algorithm, boasting a 99% success rate and offering flexibility in parameter configuration.

Looking ahead, future work will involve the comprehensive testing of additional PV parameters and diverse outdoor conditions to further enhance system efficiency. The refinement and analysis of the PV modeling approach will be prioritized to ensure the accuracy of PV error handling. Exploring alternative PV modeling techniques and conducting user satisfaction research will contribute to continual improvements, ensuring the system's effectiveness and enhancing the user experience.

Author Contributions: Conceptualization, A.D.; Methodology, A.D., A.P., K.G. and D.T. (Dimitris Triantafyllidis); Software, A.D., A.P., D.T. (Dimitris Triantafyllidis) and I.T.; Validation, A.D., A.P., K.G., D.T. (Dimitris Triantafyllidis), I.T., C.-N.A., S.K., D.I. and D.T. (Dimitrios Tzovaras); Investigation, A.D. and A.P.; Resources, S.K.; Data curation, K.G., D.T. (Dimitris Triantafyllidis), I.T., C.-N.A., S.K., D.I. and D.T. (Dimitrios Tzovaras); Writing–original draft, A.D., A.P., K.G., D.T. (Dimitris Triantafyllidis), I.T., C.-N.A., S.K., D.I. and D.T. (Dimitrios Tzovaras); Writing–review & editing, A.D., A.P., K.G., D.T. (Dimitris Triantafyllidis), I.T., C.-N.A., S.K., D.I. and D.T. (Dimitrios Tzovaras); Visualization, K.G., D.T. (Dimitris Triantafyllidis), I.T., C.-N.A., S.K., D.I. and D.T. (Dimitrios Tzovaras); Supervision,

References

1. Kingsley-Amaehule, M.; Uhunmwangho, R.; Nwazor, N.; Okedu, K.E. Smart Intelligent Monitoring and Maintenance Management of Photo-voltaic Systems. *Int. J. Smart Grid* **2022**, *6*, 110–122.
2. Jaiswal, S.P.; Shrivastava, V.; Palwalia, D.K. Opportunities and challenges of PV technology in power system. *Mater. Today Proc.* **2021**, *34*, 593–597. [CrossRef]
3. Rasheed, M.; Shihab, S.; Rashid, T.; Ounis, T.D. Parameters Determination of PV Cell Using Computation Methods. *J. Al-Qadisiyah Comput. Sci. Math.* **2021**, *13*, 1. [CrossRef]
4. Limouni, T.; Yaagoubi, R.; Bouziane, K.; Guissi, K.; Baali, E.H. Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model. *Renew. Energy* **2023**, *205*, 1010–1024. [CrossRef]
5. Rasheed, M.; Alabdali, O.; Shihab, S. A New Technique for Solar Cell Parameters Estimation of The Single-Diode Model. *J. Phys. Conf. Ser.* **2021**, *1879*, 032120. [CrossRef]
6. Nayak, B.; Mohapatra, A.; Mohanty, K.B. Parameter estimation of single diode PV module based on GWO algorithm. *Renew. Energy Focus* **2019**, *30*, 1–12. [CrossRef]
7. Rasheed, M.; Shihab, S.; Rashid, T. The Single Diode Model for PV Characteristics Using Electrical Circuit. *J. Al-Qadisiyah Comput. Sci. Math.* **2021**, *13*, 131. [CrossRef]
8. Mazzeo, D.; Matera, N.; De Luca, P.; Baglivo, C.; Congedo, P.M.; Oliveti, G. A literature review and statistical analysis of photovoltaic-wind hybrid renewable system research by considering the most relevant 550 articles: An upgradable matrix literature database. *J. Clean. Prod.* **2021**, *295*, 126070. [CrossRef]
9. Rasheed, M.; Alabdali, O.; Hassan, H.H. Parameters Extraction of a Single-Diode Model of Photovoltaic Cell Using False Position Iterative Method. *J. Phys. Conf. Ser.* **2021**, *1879*, 032113. [CrossRef]
10. Houssein, E.H.; Zaki, G.N.; Diab, A.A.Z.; Younis, E.M. An efficient Manta Ray Foraging Optimization algorithm for parameter extraction of three-diode photovoltaic model. *Comput. Electr. Eng.* **2021**, *94*, 107304. [CrossRef]
11. Korkas, C.; Dimara, A.; Michailidis, I.; Krinidis, S.; Marin-Perez, R.; Martínez, García, A.I.; Skarmeta, A.; Kitsikoudis, K.; Kosmatopoulos, E.; Anagnostopoulos, C.N.; et al. Integration and Verification of PLUG-N-HARVEST ICT Platform for Intelligent Management of Buildings. *Energies* **2022**, *15*, 2610. [CrossRef]
12. Ayadi, F.; Colak, I.; Genc, N.; Bulbul, H.I. Impacts of wind speed and humidity on the performance of photovoltaic module. In Proceedings of the 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA), Brasov, Romania, 3–6 November 2019; IEEE: Piscataway, NJ, USA, 2019.
13. Kim, G.G.; Choi, J.H.; Park, S.Y.; Bhang, B.G.; Nam, W.J.; Cha, H.L.; Park, N.; Ahn, H.K. Prediction model for PV performance with correlation analysis of environmental variables. *IEEE J. Photovoltaics* **2019**, *9*, 832–841. [CrossRef]
14. Lopez-Santos, O.; Garcia, G.; Martinez-Salamero, L.; Giral, R.; Vidal-Idiarte, E.; Merchan-Riveros, M.C.; Moreno-Guzman, Y. Analysis, design, and implementation of a static conductance-based MPPT method. *IEEE Trans. Power Electron.* **2018**, *34*, 1960–1979. [CrossRef]
15. Ebrahimi, S.M.; Salahshour, E.; Malekzadeh, M.; Gordillo, F. Parameters identification of PV solar cells and modules using flexible particle swarm optimization algorithm. *Energy* **2019**, *179*, 358–372. [CrossRef]
16. Leva, S.; Mussetta, M.; Ogliari, E. PV module fault diagnosis based on microconverters and day-ahead forecast. *IEEE Trans. Ind. Electron.* **2018**, *66*, 3928–3937. [CrossRef]
17. Li, G. Analysis of errors in model parameters for photovoltaic panel. *Aust. J. Electr. Electron. Eng.* **2019**, *16*, 281–288. [CrossRef]
18. Arévalo, P.; Benavides, D.; Tostado-Véliz, M.; Aguado, J.A.; Jurado, F. Smart monitoring method for photovoltaic systems and failure control based on power smoothing techniques. *Renew. Energy* **2023**, *205*, 366–383. [CrossRef]
19. Chen, Q.; Li, X.; Zhang, Z.; Zhou, C.; Guo, Z.; Liu, Z.; Zhang, H. Remote sensing of photovoltaic scenarios: Techniques, applications and future directions. *Appl. Energy* **2023**, *333*, 120579. [CrossRef]

20. Joseph, E.; Vijaya Kumar, P.M.; Mahinder Singh, B.S.; Ching, D.L.C. Performance Monitoring Algorithm for Detection of Encapsulation Failures and Cell Corrosion in PV Modules. *Energies* **2023**, *16*, 3391. [CrossRef]
21. Yin, J.; Wang, Y.; Wang, W.; Gao, W. Experimental Verification of Kirchoff First-law in the Manipulation of Sub-Possonian Photocurrent. In *Laser Spectroscopy-Proceedings of the XIII International Conference*; World Scientific: Singapore, 1998.
22. Shockley, W.; Queisser, H.J. Detailed balance limit of efficiency of p-n junction solar cells. *J. Appl. Phs.* **1961**, *32*, 510–519. [CrossRef]
23. Daussy, C.; Guinet, M.; Amy-Klein, A.; Djerroud, K.; Hermier, Y.; Briaudeau, S.; Bordé, C.J.; Chardonnet, C. Direct determination of the Boltzmann constant by an optical method. *Phys. Rev. Lett.* **2007**, *98*, 250801. [CrossRef]
24. Marafona, J.; Chousal, J.A.G. A finite element model of EDM based on the Joule effect. *Int. J. Mach. Tools Manuf.* **2006**, *46*, 595–602. [CrossRef]
25. Ibrahim, H.; Anani, N. Variations of PV module parameters with irradiance and temperature. *Energy Procedia* **2017**, *134*, 276–285. [CrossRef]
26. Corless, R.M.; Gonnet, G.H.; Hare, D.E.; Jeffrey, D.J.; Knuth, D.E. On the LambertW function. *Adv. Comput. Math.* **1996**, *5*, 329–359. [CrossRef]
27. Rodríguez, J.D.B.; Ramos-Paja, C.A.; Mejía, E.F. Modeling and parameter calculation of photovoltaic fields in irregular weather conditions. *Ingeniería* **2012**, *17*, 37–48.
28. Batzelis, E.I.; Anagnostou, G.; Chakraborty, C.; Pal, B.C. Computation of the Lambert W function in photovoltaic modeling. In *Lecture Notes in Electrical Engineering*; JB Metzler: Berlin, Germany, 2020; pp. 583–595.
29. Holland, J.H. Genetic algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
30. Mirjalili, S. *Genetic Algorithm. Evolutionary Algorithms and Neural Networks*; Springer: Cham, Switzerland, 2019; pp. 43–55.
31. Mitchell, M.; Crutchfield, J.P.; Das, R. Evolving cellular automata with genetic algorithms: A review of recent work. In Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA'96), Moscow, Russia, 20–22 May 1996; Volume 8.
32. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2020**, *80*, 8091–8126. [CrossRef] [PubMed]
33. Beligiannis, G.N.; Moschopoulos, C.; Likothanassis, S.D. A genetic algorithm approach to school timetabling. *J. Oper. Res. Soc.* **2009**, *60*, 23–42. [CrossRef]
34. Blickle, T. Tournament selection. *Evol. Comput.* **2000**, *1*, 181–186.
35. Spears, W.M. Adapting crossover in evolutionary algorithms. In *Evolutionary Programming*; MIT Press: Cambridge, MA, USA, 1995.
36. Manjhi, S.K.; Kumar, R. Performance assessment of K-type, E-type and J-type coaxial thermocouples on the solar light beam for short duration transient measurements. *Measurement* **2019**, *146*, 343–355. [CrossRef]
37. Panambur, K.S.; Desai, S.; Singh, A.K.; Thoben, K. A Hybrid Approach for Digital Representation of Sensors in Real- Time Applications. *Procedia Manuf.* **2020**, *52*, 14–19. [CrossRef]
38. Stone, M. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. Ser. B Methodol.* **1974**, *36*, 111–133 [CrossRef]
39. Dimara, A.; Anagnostopoulos, C.; Krinidis, S.; Tzovaras, D. Personalized thermal comfort modeling through genetic algorithm. *Energy Sources Part A Recover. Util. Environ. Eff.* **2021**, *4*, 1–22. [CrossRef]
40. Lambora, A.; Gupta, K.; Chopra, K.Genetic algorithm-A literature review. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; IEEE: Piscataway, NJ, USA, 2019.
41. Hassanat, A.; Almohammadi, K.; Alkafaween, E.A.; Abunawas, E.; Hammouri, A.; Prasath, V.S. Choosing mutation and crossover ratios for genetic algorithms—A review with a new dynamic approach. *Information* **2019**, *10*, 390. [CrossRef]
42. Wendorff, A.; Botero, E.; Alonso, J.J. Comparing Different Off-the-Shelf Optimizers' Performance in Conceptual Aircraft Design. In Proceedings of the 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Washington, DC, USA, 13–17 June 2016.
43. Warsito, B.; Yasin, H.; Prahutama, A. Particle swarm optimization versus gradient based methods in optimizing neural network. *J. Phys. Conf. Ser.* **2019**, *1217*, 012101. [CrossRef]
44. Fu, Z.; Liu, G.; Guo, L. Sequential quadratic programming method for nonlinear least squares estimation and its application. *Math. Probl. Eng.* **2019**, *2019*, 3087949. [CrossRef]
45. Gressling, T. 100 Quantum machine learning (QAI). In *Data Science in Chemistry*; De Gruyter: Berlin, Germany, 2020; pp. 504–510.
46. Rao, R. Rao algorithms: Three metaphor-less simple algorithms for solving optimization problems. *Int. J. Ind. Eng. Comput.* **2020**, *11*, 107–130. [CrossRef]
47. Shadm, M.B.; Balog, R.S.; Abu-Rub, H. Model predictive control of PV sources in a smart DC distribution system: Maximum power point tracking and droop control. *IEEE Trans. Energy Convers.* **2014**, *29*, 913–921. [CrossRef]