



## Article

# A Novel Supercapacitor Model Parameters Identification Method Using Metaheuristic Gradient-Based Optimization Algorithms

Ahmad Yasin <sup>1</sup>, Rached Dhaouadi <sup>2,\*</sup>  and Shayok Mukhopadhyay <sup>3</sup> 

<sup>1</sup> Mechatronics Graduate Program, College of Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates; b00073346@aus.edu

<sup>2</sup> Department of Electrical Engineering, College of Engineering, American University of Sharjah, Sharjah 26666, United Arab Emirates

<sup>3</sup> Electrical & Computer Engineering and Computer Science Department, University of New Haven, West Haven, CT 06516, USA; smukhopadhyay@newhaven.edu

\* Correspondence: rdhaouadi@aus.edu

**Abstract:** This paper addresses the critical role of supercapacitors as energy storage systems with a specific focus on their modeling and identification. The lack of a standardized and efficient method for identifying supercapacitor parameters has a definite effect on widespread adoption of supercapacitors, especially in high-power density applications like electric vehicle regenerative braking. The study focuses on parameterizing the Zubieta model for supercapacitors, which involves identifying seven parameters using a hybrid metaheuristic gradient-based optimization (MGBO) approach. The effectiveness of the MGBO method is compared to the existing particle swarm optimization (PSO) and to the following algorithms proposed and developed in this work: ‘modified MGBO’ (M-MGBO) and two PSO variations—one combining PSO and M-MGBO and the other incorporating a local escaping operator (LCEO) with PSO. Metaheuristic- and gradient-based algorithms are both affected by problems associated with locally optimal results and with issues related to enforcing constraints/boundaries on solution values. This work develops the above-mentioned innovations to the MGBO and PSO algorithms for addressing such issues. Rigorous experimentation considering various types of input excitation provides results indicating that hybrid PSO-MGBO and PSO-LCEO outperform traditional PSO, showing improvements of 51% and 94%, respectively, while remaining comparable to M-MGBO. These hybrid approaches effectively estimate Zubieta model parameters. The findings highlight the potential of hybrid optimization strategies in enhancing precision and effectiveness in supercapacitor model parameterization.

**Keywords:** supercapacitor; Zubieta model; energy storage; particle swarm optimization; gradient-based optimization; metaheuristic algorithms; local escaping operator; parameter identification



**Citation:** Yasin, A.; Dhaouadi, R.; Mukhopadhyay, S. A Novel Supercapacitor Model Parameters Identification Method Using Metaheuristic Gradient-Based Optimization Algorithms. *Energies* **2024**, *17*, 1500. <https://doi.org/10.3390/en17061500>

Academic Editor: Dinko Vukadinović

Received: 20 January 2024

Revised: 6 March 2024

Accepted: 9 March 2024

Published: 21 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Recently, there has been a noteworthy global upswing in the adoption of renewable energy sources. This increased interest is a proactive response to economic, political, and social factors, all directed toward diminishing dependence on conventional fossil fuels [1,2]. The efficiency of power systems that encompass a substantial share of renewable energy sources is shaped by various factors. Challenges arise from the inherent unpredictability associated with renewable energy sources. Consequently, the use of energy storage systems plays a pivotal role in effectively bringing renewable energy systems into commercial viability [3].

In recent years, there has been a marked uptick in interest surrounding supercapacitors (SCs). They are being viewed as a viable supplementary power source owing to their outstanding high power density and relatively high energy density [4]. The successful incorporation of supercapacitors into energy storage systems has been observed across diverse industrial applications, such as electric vehicles and solar energy systems [5,6]. The

integration of SCs provides significant advantages in maintaining stability within electrical power systems. This is achieved by augmenting the energy supply derived from batteries and the unpredictable nature of renewable resources.

To optimize the integration of supercapacitors into energy storage systems, it is crucial to establish a precise dynamic representation that effectively represents the static and dynamic properties of supercapacitors. The precise definition and modeling of the system's characteristics is a vital step in enhancing and managing any energy storage system. SC dynamic modeling is used to identify and characterize electrical and thermal performances, condition diagnostics-monitoring and estimation of state of charge (SOC), state of power (SOP), state of health (SOH), and control mechanism design [7–9]. Supercapacitors are modeled based on the characteristics to be monitored, with five key models: equivalent circuit models, electrochemical models, thermal models, fractional order models, and intelligent models [7,10].

Different models have been developed to explain the behavior of supercapacitors; however, the electrical equivalent circuit models are found to be a convenient and common way to simulate the electrical behavior of SCs. To characterize and simulate the electrical behavior of SC, equivalent circuit models utilize parameterized RC networks defined by ODEs. Utilizing electrical models enables the assessment of supercapacitor capacitance, considering its variations with bias voltage, voltage drop, as well as power loss attributed to internal resistance, self-discharge, and leakage current effects. Additionally, these models account for the electric dynamic behavior influenced by ion diffusion [11,12].

Modeling and characterization of supercapacitors have been extensively investigated using electrochemical impedance spectroscopy (EIS) approaches [13–16] or time response experiments and simulations [17–20]. Both techniques aim to provide a variety of equivalent circuit representations that are required for characterizing the state of the SC cell in the course of operation. Hence, when analyzing the time domain system, various circuit models have been proposed. Classical models, multi-stage ladder models, and dynamic models are the three categories of equivalent circuit models. In addition, a wide range of nonlinear models have been formulated for supercapacitors, which include fractional order models, as well as methodologies for determining parameters, thereby demonstrating the effectiveness of these models [21,22]. Nevertheless, the procedure of determining parameters for these models sometimes entails multiple stages of linearization and extensive experimental investigation.

Methods such as adaptive filter algorithms, metaheuristic optimization methods, and intelligent artificial intelligence are being deployed in the literature to find the optimum parameters for the different SC equivalent models. The recursive least squares (RLS) method was used to identify the parameters of the classical equivalent circuit model in [23–25]. In [25], the authors used a voltage-dependent capacitor in the first branch instead of the constant capacitance capacitor, making it simpler to identify the time-variant parameters. Similarly, in [26], a nonlinear least square method was used to identify the parameters of a supercapacitor model. The authors provided a framework for the branch number selection criteria and order reduction. The total least-squares method was used in [27] to identify the three parameters ( $C_0$ ,  $K_V$ , and  $C_2$ ) of the reduced two-branch Zubieta model [28]. The first branch resistor,  $R_0$ , was assumed to be given by the manufacturer, and the second branch resistor,  $R_2$ , and the leakage resistor,  $R_{lea}$ , were found using the circuit analysis method. The voltage response, input current, and their second- and first-order derivatives were used to define the estimation problem. The derivatives of the electric variables were obtained using filter differentiation.

Moreover, Kalman filters were used to identify supercapacitor dynamic model parameters online [29]. The authors used the extended Kalman filter to automate the estimation of the error bound. The researchers proved that the proposed estimator model can be used to accurately depict the voltage behavior of the SC under various scenarios.

Optimization is still a challenging computational task. Consequently, numerous algorithms have been proposed to address this challenge. Two questions must be answered to

ensure the best possible solution to this problem: how to identify global and local optimization and how to preserve such optimization until the end of the search. Swarm intelligence, which includes particle swarm optimization (PSO) [30], the grey wolf optimizer (GWO) [31], whale optimizer [32], ant-lion optimizer (ALO) [33], genetic algorithm (GA) [34], artificial ecosystem-based optimizer (AEO) [35], and many others, evolved as a result of these questions over the last two decades. These metaheuristic algorithms are widely used to solve and optimize various real-world problems. However, few researchers in the field of energy storage have used metaheuristic optimization algorithms as an appealing tool for identifying electrical circuit model parameters.

In 2022, a study by researchers [36] employed a straightforward and reliable methodology, using the bald eagle search (BES) optimization algorithm to identify the parameters of the Zubieta model's supercapacitor equivalent circuit [20]. The envisaged bald eagle search algorithm mimics bald eagle hunting behavior to demonstrate the consequences of each stage of hunting. The BES approach's robustness was assessed in comparison to other metaheuristic algorithms for two supercapacitors (SCs) with modules of 470 F and 1500 F. The authors attained an MSSE of  $1.32 \times 10^{-8}$  for the 1500 F SC when employing the BES. In contrast, the PSO, the most closely related optimizer based on the obtained results, yielded a higher MSSE of  $6.69 \times 10^{-6}$ .

In [37], a similar approach was used with the interior search optimizer (ISA), a notable metaheuristic optimization algorithm proposed by Gandomi [38]. The mean square error (MSE) for parameterization of a 470 F capacitor using the ISA was 0.004487% compared to 0.0310895 for GWO, 0.0045% for GA, and 0.03109% for WA. These findings suggest that the ISA method can be used to optimize the parameters of the Zubieta model, and they are comparable to the genetic algorithm approach.

In [39], a real-time modeling approach based on the weighting bat algorithm (WBA) was proposed. The model was used for parameter identification in the reduced Zubieta model, enabling real-time power management in embedded systems powered by supercapacitors, as well as predicting supercapacitor behavior. The WBA convergence of the fitness function in 50 iterations is comparable to the genetic algorithm with lower computational time, 0.4328% and 0.43815%, respectively.

Furthermore, the study in [40] introduced an electrical model that is based on the optimization of parameters for a passive electrical circuit model. There exist multiple methodologies for the parameter identification of supercapacitor models. These include the analytical method [20], Segmentation Optimization [41], the binary quadratic equation fitting method [42], Universal Adaptive Stabilization and Optimization [11], Particle Swarm Algorithm (PSO) [43], the recursive least-squares method [44], and other alternative techniques.

Over the past few years, there has been a notable influx of research focusing on the development and application of advanced optimizers for parameter identification, particularly in the field of energy storage systems. Metaheuristic algorithms have continued to dominate the landscape of parameter estimation, offering innovative solutions to address challenges associated with accuracy and convergence rates.

One prominent optimizer in recent literature is the Sine Cosine Algorithm (SCA), introduced by Mirjalili et al. [45]. SCA has shown promise in optimizing complex and nonlinear functions, making it suitable for parameter identification tasks. Studies have applied SCA to various energy storage system models, demonstrating its effectiveness in achieving accurate parameter estimates within a reduced computational time.

Another development, as mentioned earlier, is the grey wolf optimizer (GWO), proposed by Mirjalili et al. [31]. GWO draws inspiration from the social hierarchy of grey wolves and has been successfully applied to parameter identification tasks in renewable energy systems. Its ability to strike a balance between exploitation and exploration makes it well-suited for handling complex and high-dimensional optimization problems.

Other noteworthy contributions are the Dandelion Optimization Algorithm [46] used for the parameter estimation of proton exchange membrane fuel cells (PEMFCs) and the

Metaheuristic Mountain Gazelle Optimizer for parameter estimation of single- and double-diode photovoltaic cell models [47]. These new optimization strategies demonstrated improved performance compared with traditional algorithms, showcasing the potential of metaheuristic optimization paradigms to address the complexities associated with renewable energy system models.

Nevertheless, the existing literature points out limitations in the application of metaheuristic optimizers to identify variables in the SCs equivalent electrical circuit model. Additionally, the currently employed algorithms have their own constraints and highlight notable gaps that warrant further investigation. Existing studies often lack a standardized approach, leading to inconsistencies in parameter estimation. Challenges include the absence of a universally accepted model and the need for improved convergence rates and accuracy in optimization algorithms. Moreover, there is a limited exploration of hybrid optimization strategies that could potentially enhance the efficiency of parameter identification for supercapacitors. Addressing these gaps is crucial to advancing the reliability and performance of supercapacitors in energy storage systems, promoting a more standardized and effective methodology for parameter identification. Consequently, to build an accurate model for SCs, it is vital to use an efficient optimization technique that effectively addresses the limitations of current optimizers.

In this study, the authors propose a more straightforward and reliable approach by employing the metaheuristic gradient-based optimization (MGBO) algorithm [48,49] to ascertain the parameters of the SCs electrical circuit model, specifically the Zubieta model. Having a precise model is crucial for accurately characterizing the behavior of SCs, thereby facilitating further research in the field. Moreover, to enhance the effectiveness of PSO and address the underlying constraints, numerous variations of PSO have been developed. Multiple implementations of PSO algorithm variants have been proposed in the literature. These variants include adaptive inertia weight PSO, constriction PSO, and hybrid PSO [50,51]. These changes encompass the incorporation of additional strategies, such as the dynamic manipulation of the inertia weight, the self-adjustment of acceleration constants, and integrating it with other optimization techniques [52].

Hence, this research develops a modified metaheuristic gradient-based optimization (M-MGBO) algorithm and two versions of the particle swarm optimization (PSO) algorithm. The motivation for doing this is as follows:

Gradient-based algorithms such as the MGBO can suffer from problems related to computing/approximating the gradient. Particle swarm optimization (PSO)/metaheuristic approaches have been used to circumvent issues related to the gradient. Both MGBO and PSO approaches may be affected due to solutions being trapped in local optima. The literature has several randomized approaches to avoid such issues. However, such randomized approaches often disregard/do not explicitly incorporate checking for parameter bounds and also do not necessarily check if a randomized candidate solution actually helped achieve the objective of not being trapped in a local optimum. Furthermore, it is possible that due to the nature of the optimization problem or the algorithm or both, such issues related to entrapment in local optima, and parameter-bound constraint violation happen more frequently during a certain stage (early iterations/late iterations) of the iterative optimization cycle. Moreover, encoding such information in the optimization algorithm itself may help achieve better results.

In addition to improving parameter estimation accuracy, which requires a reduced number of algorithm iterations, alleviating the above-mentioned issues is the objective of this research. To alleviate the above issues, this work develops the M-MGBO algorithm, which incorporates an appropriate local escaping operator (LCEO) algorithm, the details of which are available in Section 3.3.1. Further, results exist in the literature showing the benefits of generating hybrid versions of the PSO algorithm [53]. Therefore, this work combines the MBGO algorithm with the PSO and also further combines the advanced LCEO algorithm mentioned earlier with the PSO algorithm, using them for supercapacitor parameter estimation. The relevant details are available in Sections 3.3.2 and 3.3.3.

## 2. Supercapacitor (SC) Model and Experimental Setup

### 2.1. SC Model

Various models have been devised to describe the behavior of supercapacitors. Nevertheless, electrical equivalent circuit models have emerged as a practical and prevalent method for simulating the electrical characteristics of supercapacitors (SCs). In characterizing and simulating the electrical behavior of an SC, equivalent circuit models use parameterized RC networks. These electrical models serve to determine the SC capacitance and its correlation with voltage drop, bias voltage, self-discharge, power loss due to internal resistance, leakage current effects, and the electric dynamic behavior induced by ion diffusion [1,2].

Furthermore, the SC model can be conceptualized as an electrochemical double-layer capacitor (DLC) [3]. An exemplary model proposed by [4] is the Zubieta model, which, through a comprehensive consideration of the device's behaviors under various operating conditions and states, has demonstrated an accurate reflection of the physics of the SC. The Zubieta model comprises three parallel RC branches with a leakage resistance (EPR), as illustrated in Figure 1.

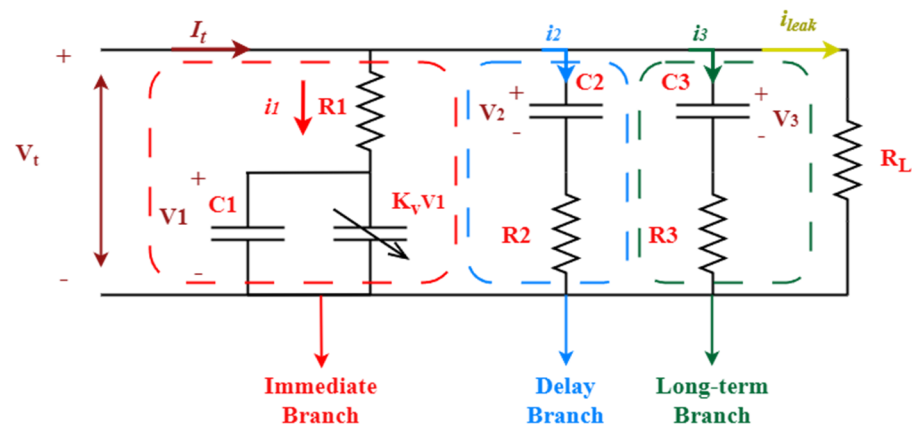


Figure 1. Supercapacitor Zubieta model.

Each branch is characterized by a unique time constant indicative of SC behavior across a specific frequency range. The Zubieta model includes three RC branches, namely the immediate branch, the delayed branch, and the long-term branch. The immediate (first) branch is characterized by a voltage-dependent differential capacitance with a constant capacitor ( $C_1$ ) and a parallel voltage-dependent capacitor ( $K_v \times V_1$ ). This first branch significantly influences the supercapacitor's behavior during the initial transient time of operation. The second and third branches (delayed branch and long-term branch) represent SC behavior over a longer period of time (minutes and hours, respectively). The equivalent series resistance (EPR) is denoted by  $R_L$ .

The incorporation of three RC branches allows the model to capture SC characteristics on distinct timescales, with the immediate branch providing a nonlinear response emulating the effects of the internal porous double-layer electrode structure of the supercapacitor. The model parameters are determined using voltage–current measurements, ensuring an accurate representation of SC behavior. This model is a valuable tool for optimizing SC performance and improving energy storage systems.

Referring to Figure 1, the total capacitance of the first branch can be calculated as follows:

$$C_0 = C_1 + \frac{K_v}{2} \times V_1(t), \quad (1)$$

where  $C_1$  represents the electrostatic capacitance of the capacitor and  $K_v$  represents the effects of the diffused layer of the SC, and it is the slope of the capacitance as a function of voltage in  $F/V$ .

The dynamics of the immediate branch can be found as follows

$$\frac{dV_1}{dt} = \frac{V_t(t) - V_1(t)}{\left(C_1 + \frac{K_v}{2} \cdot V_1(t)\right) \cdot R_1} \tag{2}$$

Similarly, the dynamic of the second and third branches (delayed branch and long-term branch) are given by the following:

$$\frac{dV_2}{dt} = \frac{V_t(t) - V_2(t)}{C_2 \cdot R_2} \tag{3}$$

$$\frac{dV_3}{dt} = \frac{V_t(t) - V_3(t)}{C_3 \cdot R_3} \tag{4}$$

The current resulting from self-discharge can be represented using the following equation:

$$i_{leak} = \frac{V_t}{R_L} \tag{5}$$

Hence, the current input to the supercapacitor can be expressed as the summation of all currents passing through all four branches in the Zubieta model representation,  $I_t = i_1 + i_2 + i_3 + i_{leak}$ . Computing the currents in terms of the voltages, further allows the development of a mathematical relation that expresses the SC's terminal voltage concerning all branches as follows:

$$V_t \times \left(1 + \frac{R_1}{R_2} + \frac{R_1}{R_3} + \frac{R_1}{R_L}\right) = V_1 + \left(\frac{R_1}{R_2}\right)V_2 + \left(\frac{R_1}{R_3}\right)V_3 + I_t R_1 \tag{6}$$

By rearranging (6), the following relation is obtained for the SC terminal voltage:

$$V_t = \left(\frac{R_2 R_3 R_L}{R_2 R_3 R_L + R_2 R_1 R_L + R_1 R_3 R_L + R_1 R_2 R_3}\right) \left(V_1 + \left(\frac{R_1}{R_2}\right)V_2 + \left(\frac{R_1}{R_3}\right)V_3 + I_t R_1\right) \tag{7}$$

Hence, employing the derived dynamics for the three branches and having knowledge of the input/total current, the Simulink representation depicted in Figure 2 is employed. This approach enables the calculation of the estimated terminal voltage once the electrical parameters of the supercapacitor (SC) are identified. Subsequently, the computed terminal voltage is cross-referenced with the measured terminal voltage, acting as a validation step to verify the accuracy of the obtained model.

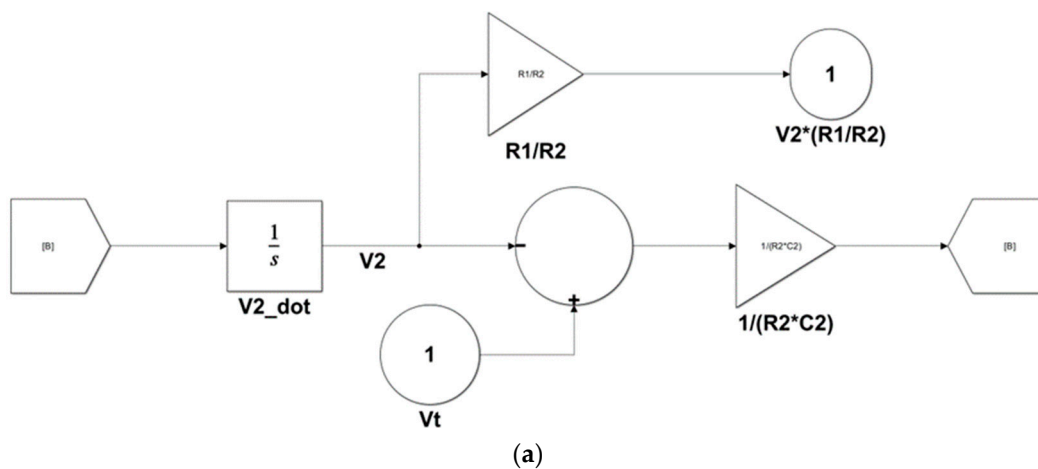
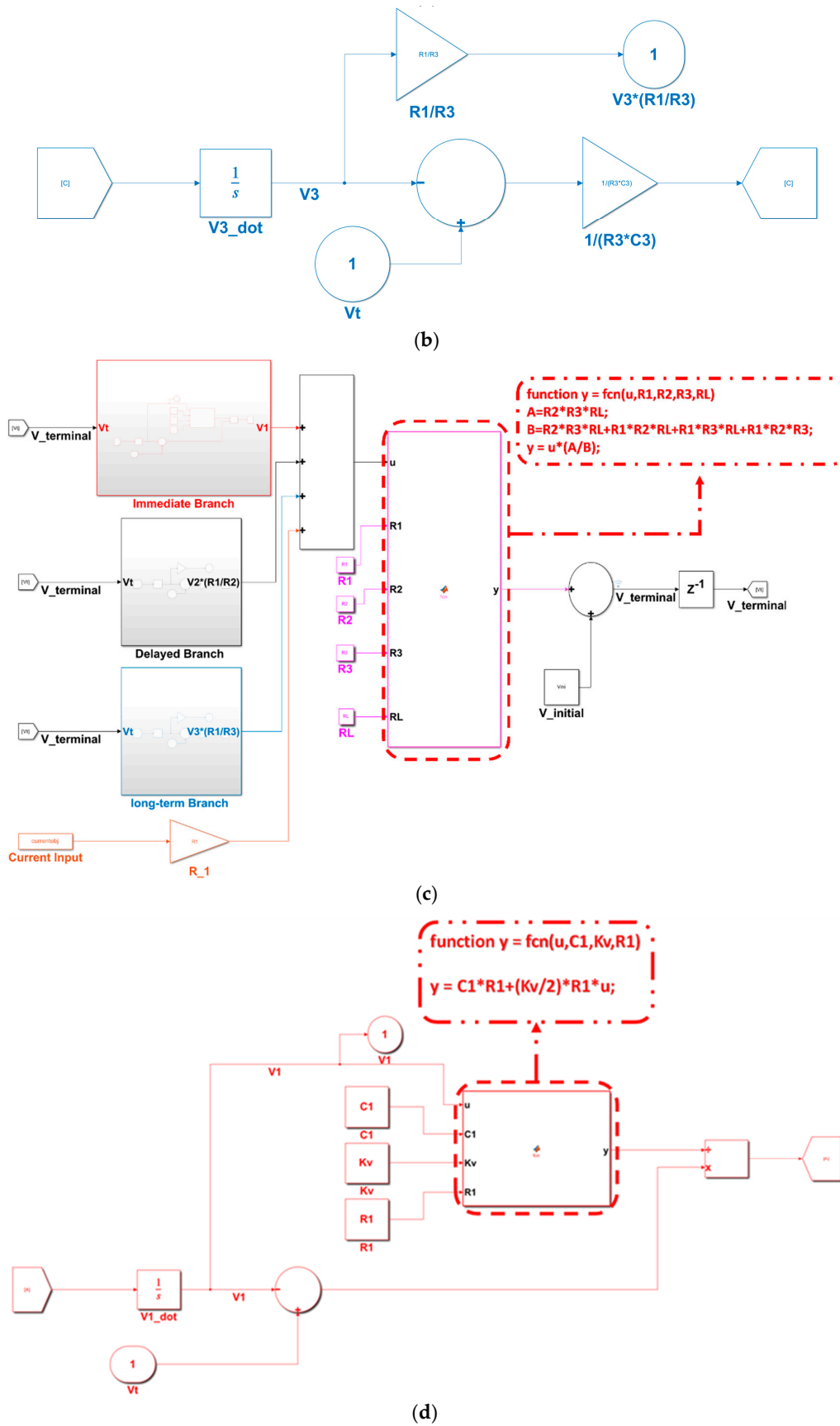


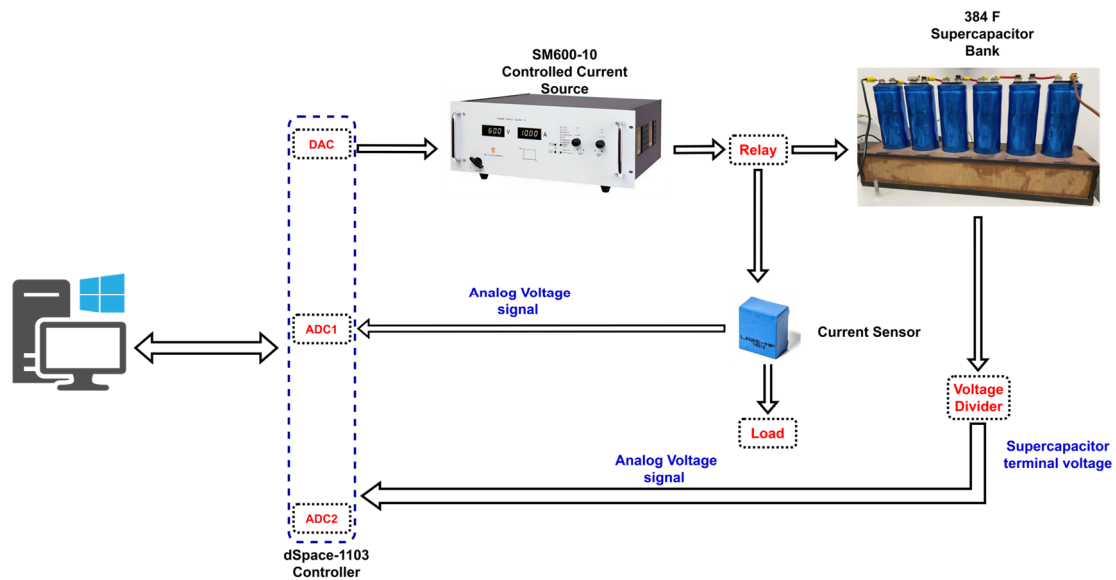
Figure 2. Cont.



**Figure 2.** Supercapacitor Zubieta model Simulink representation: (a) long-term branch; (b) delayed branch; (c) main model; (d) immediate branch.

## 2.2. Experimental Setup

The experimental setup configuration comprises a supercapacitor bank, a controlled current source, current and voltage sensors, and a dSPACE controller, as illustrated in Figure 3. The controlled current source was modulated using an analog signal through dSPACE. A voltage signal ranging from 0 V to 5 V governed the remote control of the power supply. The interface of the power supply with the dSPACE controller is depicted in Figure 3. The dSPACE controller monitored the terminal voltage of the supercapacitor bank, and the charging current was measured using the LA25-NP current transducer. To ensure that the analog signal read by the dSPACE controller remains below 10 V (1 after scaling), a voltage divider was employed.

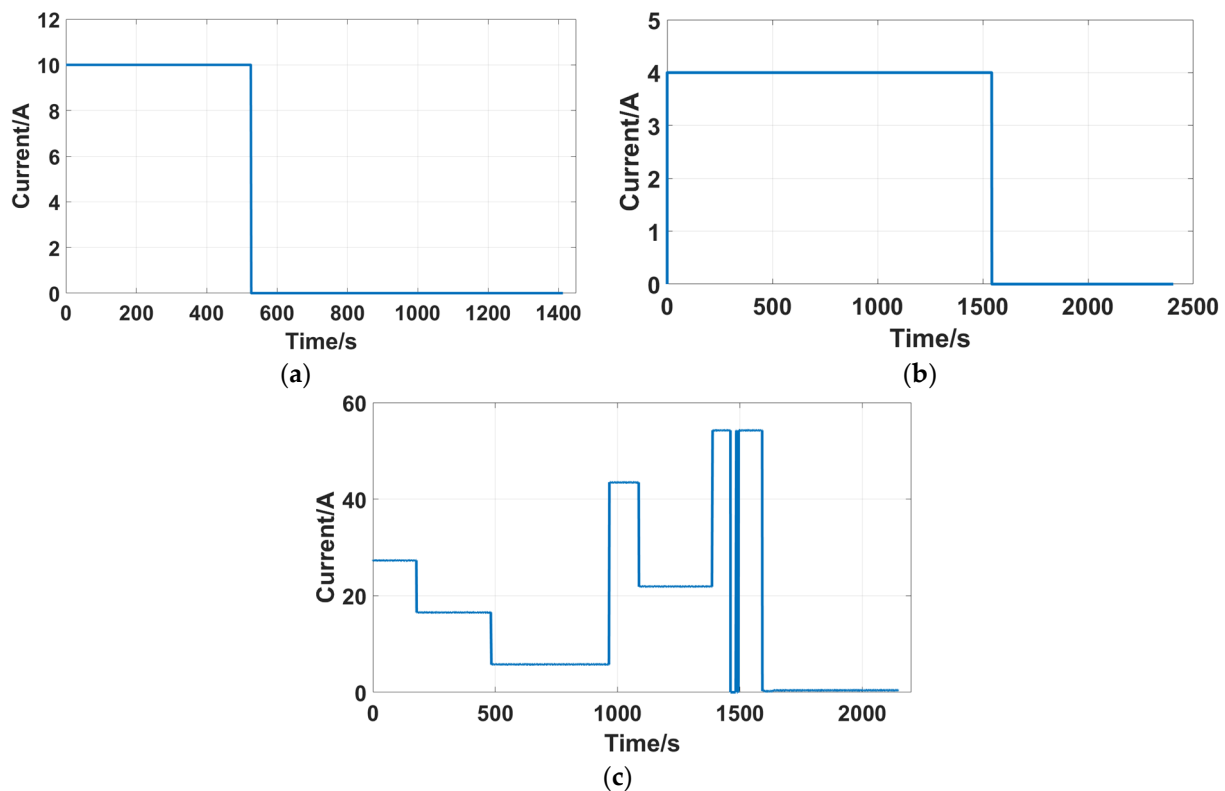


**Figure 3.** Experimental setup block diagram.

The current source was programmed to generate three distinct charging profiles: Profile 1: a 10 A step input; Profile 2: a 4 A step input; and Profile 3: a variable frequency input, as illustrated in Figure 4, respectively. For each of these input scenarios, the supercapacitor bank undergoes charging until it reaches its rated voltage of 15 V. At this point, the relay between the current source and the supercapacitor bank is deactivated, allowing the SC bank to settle for a duration of 30 min. Throughout both the charging and settling processes, the terminal voltage of the SC bank is continuously monitored by the dSPACE controller.

Following the 30 min settling period, the SC bank was discharged to a load. In this analytical study, particular emphasis was placed on the voltage response of the SC bank during the charging process as a key element in the optimization problem. The model generated was designed to emulate the behavior of the supercapacitor during the charging process.





**Figure 4.** Experimental current profiles: (a) Profile 1: 10 A step input; (b) Profile 2: 4 A step input; (c) Profile 3: Variable frequency input.

### 2.3. SC Model Validation

The determination of equivalent circuit model parameters is a crucial aspect of the optimization of SC operation, and various approaches have been proposed in the literature for this purpose. One such approach is the circuit analysis/analytical method, which establishes relationships between different circuit components, allowing for the straightforward identification of the equivalent circuit model parameters. This method has the advantage of having a clear physical interpretation and not requiring extensive experimental equipment [5].

In the case of the three-branch Zubieta model, a procedure was proposed by Zubieta et al. [4] to determine the parameters of the SC model. This method involves charging a fully discharged SC with a high constant current. The immediate branch parameters are identified by measuring the voltage response at a short time constant, as it is assumed that this branch has a smaller time constant compared with the other two branches. This results in all of the charges being initially stored in the immediate branch. Once the capacitor reaches its rated voltage, the charging current is set to zero, and the parameters of the delayed branch and long-term branch are then estimated using the voltage response at the appropriate time constants of each branch.

The parameterization procedure for the Zubieta model is summarized in Table 1. By employing this method, it is possible to reliably determine the parameters of each branch of the equivalent circuit model using the SC voltage response at the appropriate time constants.

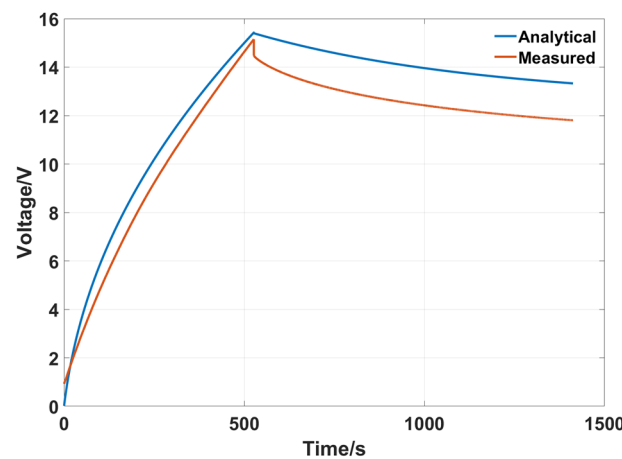
**Table 1.** Zubieta model analytical parameterization procedure.

Parameter	Relation	Comments
$R_1$	$R_1 = \frac{V_1}{I_{sc}}$	The SC is assumed to start charging at zero initial conditions. $V_1$ is the voltage at $t_1$ where $20 \text{ ms} < t_1 < 50 \text{ ms}$
$C_1$	$C_1 = \frac{i_{sc}(t_2 - t_1)}{\Delta V}$	Where $t_2$ is the time when $\Delta V = V_2 - V_1 = 50 \text{ mV}$
$K_v$	$K_v = \frac{2}{V_4} \left( \frac{Q_t}{V_4} - C_0 \right)$	Where the amount of charge stored $Q_t = i_{sc}(t_4 - t_1)$ , and $t_4 = t_3 + 20 \text{ ms}$ . Where $t_3$ is when $V_{sc} = V_{rated}$
$R_2$	$R_2 = \frac{(V_4 - \frac{\Delta V_2}{2})(t_5 - t_4)}{(C_0 + k(V_4 - \frac{\Delta V_2}{2}))\Delta V_2}$	Where $t_5$ is at $V_5 = V_4 - \Delta V_2$ . Where $\Delta V_2 = 50 \text{ mV}$
$C_2$	$C_2 = \frac{Q_t}{V_6} - \left( C_0 + \frac{kV_6}{2} \right)$	$V_6$ is the voltage at $t_6 = t_5 + 300 \text{ s}$
$R_3$	$R_3 = \frac{(V_7 - \frac{\Delta V_2}{2})(t_7 - t_6)}{(C_0 + k(V_6 - \frac{\Delta V_2}{2}))\Delta V_2}$	Where $t_7$ is at $V_7 = V_6 - \Delta V_2$ . Where $\Delta V_2 = 50 \text{ mV}$
$C_3$	$C_3 = \frac{Q_t}{V_7} - \left( C_0 + \frac{kV_7}{2} \right) - C_2$	Where $V_8$ is at $t_8 = 30 \text{ min}$

Accordingly, the procedure in Table 1 is used to find the parameters of the Zubieta model for a 384 F SC bank rated at 15 V using Profile 1 (Figure 4a). The identified parameters are presented in Table 2, and the corresponding voltage response is graphically represented in Figure 5.

**Table 2.** Identified SC parameters using Zubieta model analytical parameterization procedure.

Parameter	Value
$K_v \left( \frac{F}{V} \right)$	53.4472
$R_1 (\Omega)$	0.0041
$C_1 (F)$	0.8516
$R_2 (\Omega)$	0.0024
$C_2 (F)$	69.7146
$R_3 (\Omega)$	4.9780
$C_3 (F)$	126.8252
$T_1$	0.00349156
$T_2$	0.167315
$T_3$	631.4593



**Figure 5.** Experimental voltage response vs. the analytical response using current Profile 1.

Despite its simplicity, the analytical method is not suitable for real-life applications, as it does not provide accurate parameter estimates for the Zubieta model with inputs

other than a step input. However, the parameters obtained using this method can serve as an initial estimate for parameterization. A study by researchers in [6] compared the Zubieta model's behavior to that of a real supercapacitor. The research found that there is a discrepancy between the simulated and actual behavior due to the model's inability to account for self-discharge phenomena. Additionally, the quality of data and charging current amplitude significantly affect the parameterization process.

### 3. Proposed Identification Methods

This section introduces our proposed procedure for identifying the SC Zubieta model parameters. Each phase of this process is explained in the subsequent subsections.

#### 3.1. Metaheuristic Gradient-Based Optimization (MGBO)

The metaheuristic gradient-based optimization (MGBO) algorithm was first introduced in [7]. This method was further applied for the parameter identification of single and two-diode models of solar cells [8]. The MGBO method incorporates both gradient and population-based techniques to determine the search direction via Newton's method (Given by (8)). MGBO uses a set of vectors and two key operators, the gradient search rule and local escaping operators, to traverse the search domain and minimize the objective function in optimization problems.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \frac{2\Delta x \times f(x_n)}{f(x_n + \Delta x) - f(x_n - \Delta x)} \quad (8)$$

The MGBO algorithm applied a variant of the Newton's method in its formation proposed by [9]. The methodology followed by this variant improves the iterative process of the method by approximating the indefinite integral by a trapezoid instead of a rectangle. This reduces the error in the approximation and uses the average vector of  $z_{n+1}$  and  $x_n$ , instead of using  $x_n$  only. The enhanced representation is given by the following:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'\left(\frac{[z_{n+1} + x_n]}{2}\right)} \text{ where,} \quad (9)$$

$$z_{n+1} = x_n - \frac{f(x_n)}{(f')} \quad (10)$$

##### 3.1.1. Gradient Search Rule (GSR) and Algorithm Formulation

The formulation of the gradient search rule (GSR) is based on an approach that incorporates the use of Newton's gradient. The use of this methodology entails regulating the exploration of the solution vector within the search space with the aim of augmenting the rate at which the metaheuristic GBO achieves convergence. The GSR, using (8), is formulated as follows:

$$\text{GSR} = 2\Delta x \cdot \mu_1 \cdot r_n \frac{X_i^{it}}{x_{worse} - x_{best} + \varepsilon'} \quad (11)$$

where in (11),  $r_n$  is a normal distributed random number,  $X_i^{it}$  is the current approximation ( $x_n$ ),  $\Delta x$  is the derivative incrementation value, and  $x_{worse}$  and  $x_{best}$  are the worst and best positions neighboring  $X_i^{it}$  in the optimization process. The search capabilities of the MGBO algorithm are enhanced by adding a random parameter  $\mu_1$  to the GSR, where it is given by the following relation:

$$\mu_1 = 2 \cdot r \cdot \alpha - \alpha, \quad (12)$$

where  $r$  is a random number between 0 and 1 and

$$\alpha = \left| \beta \sin\left(\frac{3\pi}{2} + \sin\left(\frac{\beta 3\pi}{2}\right)\right) \right|, \quad (13)$$

$$\beta = \beta_{min} + (\beta_{max} - \beta_{min}) \left(1 - \left(\frac{m}{M}\right)^3\right)^2, \quad (14)$$

where  $\beta_{max}$  and  $\beta_{min}$  are 0.2 and 1.2, respectively,  $m$  is the number of the current iteration, and  $M$  is the total number of iterations. The optimization process involves searching for the optimal solution in the given search space. The parameter  $\alpha$  is introduced to enhance the algorithm's balance between local and global search.

In contrast,  $\Delta x$  is defined as the difference between the best position and a randomly selected position from the population of the current iteration.

$$\Delta x = rand(1 : N) \times |S|, \quad (15)$$

$$S = \frac{(x_{best} - x_{r1}^m) + \gamma}{2}, \quad (16)$$

$$\gamma = 2 \cdot r \left( \left| \frac{\sum_{n=1}^4 x_{r_n}^m}{4} - x_n^m \right| \right), \quad (17)$$

where  $r_a$  and  $r_b$  are random variables in the range  $[0, 1]$ .

### 3.1.2. Direction of Movement

The direction of movement (DM) in (18) is introduced into the algorithm to enhance the exploitation of the area near the solution  $X_i^{it}$ , moving the current solution vector in the direction of  $(x_{best} - X_i^{it})$ .

$$DM = rand \times \mu_2 (x_{best} - X_i^{it}) \quad (18)$$

Hence, the GSR and DM are used to update the current position vector  $(X_i^{it})$  and the following relation is derived:

$$X1_n^m = X_i^{it} - randn \cdot \mu_1 \left( \left( \frac{2\Delta x \cdot X_i^{it}}{(yp_n^m - yq_n^m + \varepsilon)} \right) \right) + rand \cdot \mu_2 (x_{best} - X_i^{it}) \quad (19)$$

Moreover, by replacing the position vector of the best solution with the current solution vector  $(X_i^{it})$ , a newly generated vector can be generated as follows:

$$X2_n^m = x_{best} - randn \cdot \mu_1 \left( \left( \frac{2\Delta x \cdot X_i^{it}}{(yp_n^m - yq_n^m + \varepsilon)} \right) \right) + rand \cdot \mu_2 (x_{r1}^m - x_{r2}^m), \quad (20)$$

where, in the given context, the variables  $yp_n^m$  and  $yq_n^m$  are two positions centered/averaged with respect to  $z_{n+1}$  and  $X_i^{it}$  and are given by the following:

$$yp_n^m = rand \cdot \left( \frac{(z_{n+1} + X_i^{it})}{2} + rand \cdot \Delta x \right) \quad (21)$$

$$yq_n^m = rand \cdot \left( \frac{(z_{n+1} + X_i^{it})}{2} - rand \cdot \Delta x \right) \quad (22)$$

Moreover, DM for  $X2_n^m$  is given as follows:

$$DM_{X2} = rand \cdot \mu_2 (x_{r1}^m - x_{r2}^m) \quad (23)$$

Hence, the GSR can be expressed as follows:

$$GSR_{X1} = GSR_{X2} = rand \times \frac{\mu_1 (2\Delta x \times X_i^{it})}{(y_p - y_q + \varepsilon)} \quad (24)$$

The solution  $X2_n^m$  is defined as a local search direction, where although it is good for local search, it is limited to global search. Moreover, the solution  $X1_n^m$  is defined for global search (exploration), but it is limited to local search. Hence, the MGBO algorithm is developed to take advantage of both solutions ( $X1_n^m$  and  $X2_n^m$ ), as well as take advantage of both exploitation and exploration. The combination of both solutions results in the following updated solution ( $X_i^{it+1}$ ), derived from the current solution  $X_i^{it}$ :

$$X_i^{it+1} = r_a(r_b X1_n^m + (1 - r_b)X2_n^m) + (1 - r_a)X3_n^m, \quad (25)$$

where

$$X3_n^m = X_i^{it} - \mu_1(X2_n^m - X1_n^m) \quad (26)$$

and  $r_a$  and  $r_b$  are two random numbers between 0 and 1.

### 3.1.3. Local Escaping Operator LCEO

The MGBO algorithm can be susceptible to getting stuck in local optima, which can limit its ability to find the global optimum. To address this issue, the local escaping operator (LCEO) has been proposed as a way to significantly change the position of the solution  $x_n$ , leading to improved search capability. The LCEO operator generates a new solution,  $X_{LCEO}$ , by combining several solutions, including the best position ( $x_{best}$ ), the solutions  $X1_n^m$  and  $X2_n^m$ , two randomly selected solutions from the population,  $x_{r1}$  and  $x_{r2}$ , and a new randomly generated solution,  $X_z$ . This approach effectively enables the MGBO algorithm to break away from local optima, exploring new regions within the solution space, thereby enhancing the likelihood of discovering a global optimum. Hence, the LCEO is defined via the following scheme (Algorithm 1) [8]:

---

#### Algorithm 1 Local ESCAPING OPERATOR (LCEO)

---

**Require:**  $\mu_1, \mu_2, X_i^{it}, x_{best}, x_{best2}, x_{best3}, x_{r1}, x_{r2}, nP, MaxIT$  and  $LC$

```

1   $LC = 4 \times LC \times (1 - LC)$ 
2  if  $rand < LC$  then
3       $k = integer(rand(1, nP))$  // Randomly generated integer;  $1 \leq k \leq nP$ 
4       $L = rand < 0.5$  // Either 1 or 0
5       $X_z = (1 - L) \times x_{best2} + L * X_i^{it}$ 
6      if  $rand < 0.5 \times (1 - \frac{it}{MaxIt})$  then
7           $X_{LCEO} = x_{best3} + \mu_1 \times (x_{best} - X_z) + \mu_2 \times (x_{r1} - x_{r2})$ 
8      else then
9           $X_{LCEO} = x_{best} + \mu_1 \times (x_{best} - X_z) + \mu_2 \times (x_{r1} - x_{r2})$ 
10     end if
11 end if

```

---

In Algorithm 1,  $nP$  is the number of members in the population,  $LC$  is a changing variable,  $x_{r1}$  and  $x_{r2}$  are two random members from the population, and  $X_i^{it}$  is the  $i^{th}$  member current solution in iteration  $it$ .

### 3.1.4. MGBO Algorithm

To start the optimization process, the first crucial step is to initialize the population. This step involves randomly generating a set of individuals that can be used to represent potential solutions to the optimization problem. Within the framework of the MGBO algorithm,  $N$  vectors are used as the population in a  $d$ -dimensional search space. In practice, the initial population is generated by selecting  $N$  points within the search space at random, as presented in the following equation:

$$X_i = X_{min} + r \times (X_{max} - X_{min}) \quad (27)$$

In (27),  $X_{max}$  and  $X_{min}$  are the upper and lower bounds, and  $r$  is a random number between 0 and 1.

The full algorithm flow proposed by [7] is presented in Algorithm 2, which provides a detailed overview of the steps involved in the optimization process. It is worth noting that the optimization process may require multiple iterations before the best solution is found. Each iteration involves updating the positions of the population in the search space based on their individual and collective behavior. Through this process, the algorithm can systematically navigate the search space, facilitating more effective exploration and convergence towards the optimal solution.

---

**Algorithm 2** Metaheuristic gradient-based optimization

---

**Require:**  $\epsilon$ ,  $nP$ ,  $nj$ ,  $LC$ ,  $MaxIt$ ,  $X_{min}$ , and  $X_{max}$

```

1   Set  $i = 1$ , and  $it = 1$ , randomly initialize the initial population  $X_{nP \times nj}$ 
2   Evaluate the objective function for the initially generated parameters
3   Update  $x_{best1}$ ,  $x_{best2}$ ,  $x_{best3}$ 
4   for  $it < MaxIt$  do
5       Define  $\alpha$  and  $\beta$  using (13) and (14)
6       while  $i \leq nP$  do
7           Define  $X_R = [x_{r1}, x_{r2}, x_{r3}, x_{r4}]$ 
8           Find  $\mu_1$ , and  $\mu_2$  using (12)
9           Evaluate  $DM_{X1}$ ,  $DM_{X2}$  and the GSR operators using (18), (23), and (24)
10          Evaluate  $X1_n^m = X_i^{it} - GSR + DM_{X1}$ 
11          Evaluate  $X2_n^m = X_i^{it} - GSR + DM_{X2}$ 
12          Evaluate  $X3_n^m$  using (26)
13          Update  $X_i^{it}$  using (25) and compute the objective function
14          if  $rand < LC$  then; Go to Algorithm 1
15              Compute  $X_{LCEO}$  and the objective function of  $X_{LCEO}$ 
16               $X_i^{it} = X_{LCEO}$ 
17          end if
18          Update  $x_{best1}$ ,  $x_{best2}$ ,  $x_{best3}$ 
19      end while
20  end for

```

---

In Algorithm 2,  $\epsilon$  is a small number ( $0 \leq \epsilon \leq 0.1$ ),  $nj$  is the number of parameters,  $X_{min}$ , and  $X_{max}$  are the parameters upper and lower bounds, and  $MaxIt$  is the maximum number of iterations.

### 3.2. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) (Figure 6) is a metaheuristic computational method that emulates the social behavior of birds flocking or fish schooling. It is a population-based optimization algorithm that was first introduced by Kennedy and Eberhart in 1995 [10]. The algorithm starts with a population of particles, each representing a potential solution to the optimization problem. Each particle has a position in the search space and a velocity that determines how it moves through the space. The particles update their positions and

velocities based on their own experience and the experience of their neighbors, with the goal of moving toward the best solution found so far by the entire population [11]. The PSO algorithm starts with a population of particles, each representing a set of parameter values. Each particle has a position in the parameter space and a velocity that determines how it moves through the space. The particles update their positions and velocities based on their own experience and the experience of their neighbors, intending to move towards the best set of parameters found so far by the entire population [10].

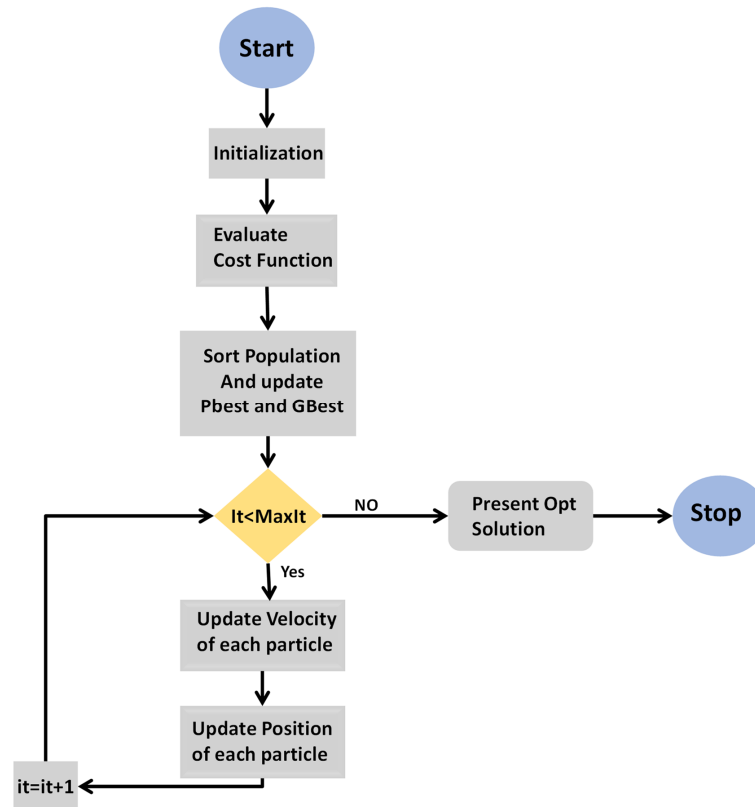


Figure 6. PSO flow chart.

The movement of a particle is determined by its current velocity, which is updated based on its personal best and global best positions, as well as a random value called the “inertia weight”. The velocity update rule is given by the following:

$$V(t+1) = W \times V(t) + c1 \times r \times (P_{best} - P(t)) + c2 \times r \times (G_{best} - P(t)), \quad (28)$$

where  $t$  is the current iteration,  $r$  is a random number between 0 and 1,  $W$  is the inertia weight,  $P_{best}$  is the best set of parameters (position values) a particle has encountered so far,  $G_{best}$  is the best set of parameters (position values) encountered by any particle in the population, and  $c1$  and  $c2$  are constants that control the relative importance of the personal best and global best positions. The position of the particle is updated based on its current velocity using the following equation:

$$Position(t+1) = Position(t) + Velocity(t+1) \quad (29)$$

### 3.3. Proposed Algorithms Variants

#### 3.3.1. Modified Metaheuristic Gradient-Based Optimization Algorithm (M-MGBO) Application for SC Parameterization

The original MGBO algorithm, as described in [7], has been augmented with certain modifications. A noteworthy addition has been the introduction of a cross-over operator that assesses the adherence of the solution generated by (25) to the upper and lower

bounds. In the MGBO algorithm, should a solution exceed these bounds, a novel solution that remains within these confines is produced for the respective population member. This approach mirrors the methodology used during the initial population generation, as presented in (27). The modified cross-over operator functions by contrasting the solution  $X_i^{it}$  of the current iteration with the upper and lower bounds. If any parameter of the solution breaches these bounds, it is substituted with its previous value from the previous iteration,  $X_i^{it-1}$ .

An additional noteworthy constraint of the original MGBO algorithm becomes apparent during its transition into the LCEO phase. In this phase, the solution obtained by the MGBO approach, marked as  $X_i^{it}$ , is replaced with the solution obtained through the LCEO ( $X_{LCEO}^i$ ). The basis for this change is grounded in the belief that the LCEO procedure often produces superior solutions, thereby enabling an escape from local minima. However, this advantage is mostly evident in the initial stages of the optimization process. The MGBO algorithm has an inherent tendency to guide the whole population towards the globally optimum solution as long as the gradient is not too small. The aforementioned tendency ultimately limits the effectiveness of the LCEO when the number of iterations increases.

Moreover, the incorporation of randomness by the LCEO algorithm serves the purpose of aiding the MGBO in converging toward the optimal solution. However, it is important to note that this randomness might also restrict the algorithm's capacity to escape from a local minimum. To tackle these concerns, several alterations have been suggested for the LCEO. The modified approach includes a comparator to evaluate and compare the responses produced by the LCEO and MGBO methods. This allows the algorithm to select the superior solution based on the fitness function. The comparator serves the additional purpose of mitigating the algorithm's tendency to become trapped in a local minimum. This is achieved by identifying instances in which the LCEO solution surpasses the MGBO solution in terms of cost function.

Secondly, the LCEO has been modified to reduce randomness and control the direction of the generated solution. The LCEO now moves towards the second and third best solutions, in addition to the best solution. This modification aims to increase population diversity and accuracy while also improving the ability of the algorithm to escape from a local minimum. By moving towards other good solutions, the LCEO is less likely to get stuck at a local minimum and more likely to find the global minimum. Moreover, to further facilitate the investigation of the LCEO technique, an additional variable denoted as  $C$  is introduced.

The value of  $C$  is used to replace  $\alpha$  in (12). The parameter  $\alpha$  is employed to determine the equilibrium between exploration and exploitation, while the variable  $C$  is incorporated to augment this dynamic within the LCEO and to increase the weight of the best solution movement. As depicted in Figure 7, the  $C$  value exhibits an increase with the decrease of  $\alpha$  (initially). In the event that the algorithm gets confined to a local minimum within the initial 100 iterations, the incorporation of the  $C$  value amplifies the LCEO's capacity to provide a diverse solution, hence augmenting its exploitative potential. As a result, a new variable is defined ( $\mu_3$ ):

$$\mu_3 = 2 \cdot r \cdot C - C \quad (30)$$

Hence, the modifications to the LCEO aim to improve the performance of the MGBO by addressing some of the limitations of the original LCEO method. By introducing a comparator and modifying the LCEO to reduce randomness and control the direction of the generated solution, the algorithm is enhanced in its ability to locate the global minimum and avoid being trapped in a local minimum.

The comprehensive pseudocode detailing the modified MGBO (M-MGBO) algorithm, inclusive of the model's constraints and the modifications, is provided in Algorithm 3. Corresponding functions associated with this algorithm can be found in Algorithm 4 and Algorithm 5.



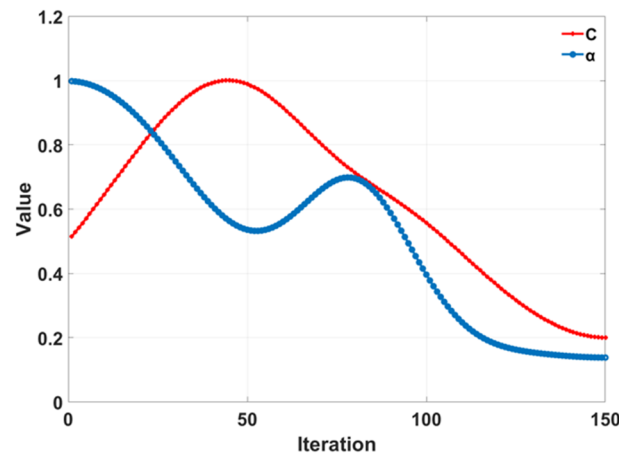


Figure 7. Comparison between  $\alpha$  and C.

The LCEO modifications have the potential to significantly enhance the performance of optimization algorithms and lead to better solutions across a broad spectrum of problems. The updated LCEO algorithm is given in Algorithm 6.

---

#### Algorithm 3 Modified MGBO

---

**Require:**  $\mathcal{E}$ ,  $nP$ ,  $nj$ ,  $LC$ ,  $MaxIt$ ,  $X_{min}$ , and  $X_{max}$

```

1  Set  $i = 1$ , and  $it = 1$ , randomly initialize the initial population  $X_{nP \times nj}$ 
2  Evaluate the MSE for the population initially;  $MSE = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ 
3  Update  $x_{best1}$ ,  $x_{best2}$ ,  $x_{best3}$ 
4  for  $it < MaxIt$  do
5      Define  $\alpha$  and  $\beta$  using (13) and (14)
6      while  $i \leq nP$  do
7           $X_{old} = X_i^{it}$  //  $X_{old} = [K_v(i), R_1(i), C_1(i), R_2(i), C_2(i), R_3(i), C_3(i), R_L(i)]_j$ 
8          Define  $X_R = [x_{r1}, x_{r2}, x_{r3}, x_{r4}]$  // Random members from the population
9          Evaluate  $\mu_1$ , and  $\mu_2$  using (12)
10         Evaluate  $DM_{X_1}$ ,  $DM_{X_2}$  and the GSR operators using (18), (23), and (24)
11         Evaluate  $X1_n^m = X_i^{it} - GSR + DM_{X_1}$ 
12         Evaluate  $X2_n^m = X_i^{it} - GSR + DM_{X_2}$ 
13         Evaluate  $X3_n^m$  using (26)
14         Update  $X_i^{it}$  using (25) //  $X_i^{it} =$ 
15          $X_i^{it} = \text{limit}(X_i^{it}, X_{old})$  // Check for upper and lower bounds, Algorithm 4
16         Let  $A = [X_i^{it}(C_1), X_i^{it}(C_2), X_i^{it}(C_3)]_j$  and  $B = [X_i^{it}(R_1), X_i^{it}(R_2), X_i^{it}(R_3)]_j$ 
17          $X_i^{it} = \text{Check}(A, B, X_{old})$  // Checking for constraints, Algorithm 5
18         Evaluate cost function of  $X_i^{it}$  using Equation in line 2.
19         if  $\text{rand} < LC$  then //Local escaping operator, Algorithm 6
20             Compute  $X_{LCEO}$  and the objective function of  $X_{LCEO}$ 
21              $X_{LCEO}^i = \text{limit}(X_{LCEO}^i, X_i^{it})$  // Check for upper and lower bounds
22             Let  $C = [X_{LCEO}^i(C_1^{LCEO}), X_{LCEO}^i(C_2^{LCEO}), X_{LCEO}^i(C_3^{LCEO})]_j$  and  $D =$ 
23              $X_{LCEO}^i = \text{Check}(C, D, X_i^{it})$  // Checking for constraints
24             Evaluate cost function for  $X_{LCEO}^j$  using Equation in line 2.
25              $X_i^{it} = \text{Compare}(X_i^{it}, X_{LE}^i, \text{Cost}(X_i^{it}), \text{Cost}(X_{LE}^i))$  // To update  $X_i^{it}$  if
26         end if
27         Update  $x_{best1}$ ,  $x_{best2}$ ,  $x_{best3}$ , and  $x_{worst}$ 
28     end while
29 end for

```

---

**Algorithm 4** Checking upper and lower boundsRequire:  $nj, X_{min}$  and  $X_{max}$ 

```

1  function  $X = limit(X1, X2)$ 
2  |   | for  $j = 1:nj$ 
3  |   |   | if  $(X1(j) < X_{min}(j) \parallel X1(j) > X_{max}(j))$  then
4  |   |   |   |  $X1(j) = X2(j)$ 
5  |   |   |   | end if
6  |   |   | end for
7  end function

```

**Algorithm 5** Comparing two members of the population and keeping the one with the lowest cost function

```

1  function  $X = compare(X1, X2, Cost1, Cost2)$ 
2  |   | if  $(Cost1 > Cost2)$  then
3  |   |   |  $X = X2$ 
4  |   |   | else
5  |   |   |   |  $X = X1$ 
6  |   |   | end if
7  end function

```

**Algorithm 6** MODIFIED Local ESCAPING OPERATOR (LCEO)Require:  $\mu_1, \mu_2, X_i^{it}, x_{best}, x_{best2}, x_{best3}, x_{r1}, x_{r2}, nP, beta, \alpha,$  and  $LC$ 

```

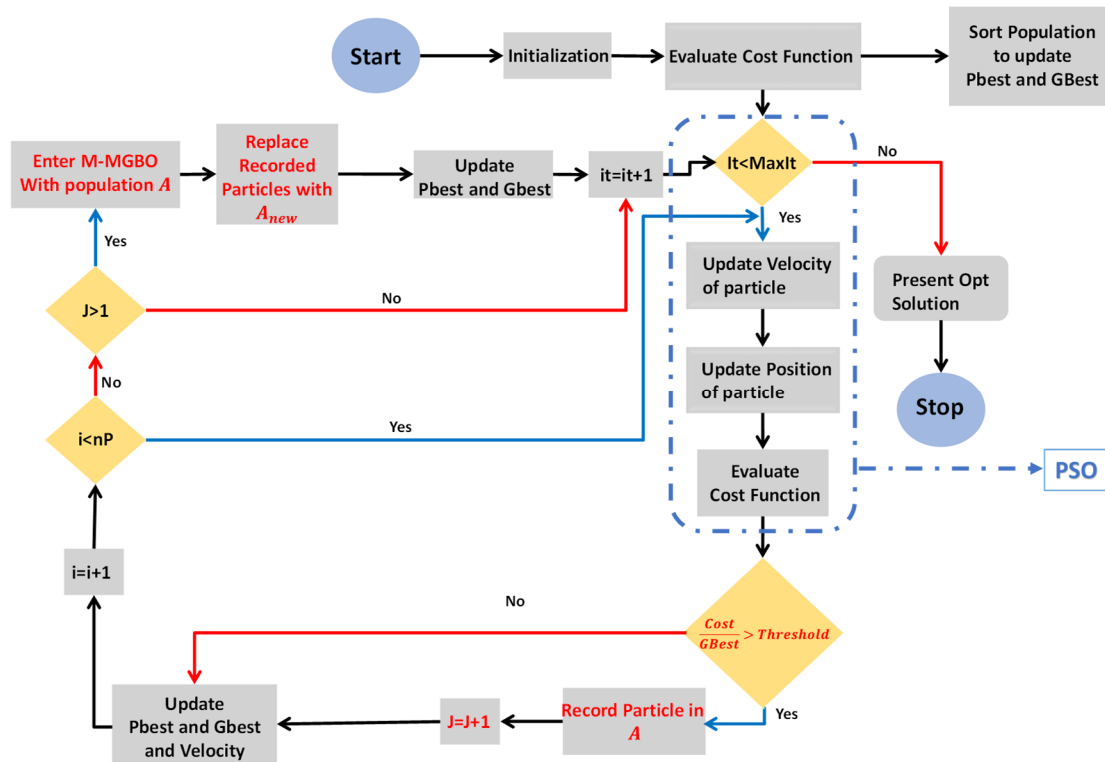
1   $LC = 4 \times LC \times (1 - LC)$ 
2  if  $rand < LC$  then
3  |   |  $k = integer(rand(1, nP))$  // Randomly generated integer;  $1 \leq k \leq nP$ 
4  |   |  $L = rand < 0.5$  // Either 1 or 0
5  |   |  $X_z = (1 - L) \times x_{best2} + L \times (x_{best3} - X_i^{it})$ 
6  |   |  $C = beta - \alpha \left(1 - \frac{it}{MaxIt}\right)^2$  and  $\mu_3 = 2 \cdot r \cdot C - C$ 
7  |   | if  $rand < 0.5 \times \left(1 - \frac{it}{MaxIt}\right)$  then
8  |   |   |  $X_{new}^{it} = x_{best2} + \mu_3 \times (x_{best} - X_z) + \mu_2 \times (x_{best3} - x_{r2})$ 
9  |   |   | else then
10 |   |   |  $X_{new}^{it} = x_{best} + \mu_3 \times (x_{best} - X_z) + \mu_2 \times (x_{best2} - x_{r1})$ 
11 |   |   | end if
12 end if

```

### 3.3.2. Hybrid PSO-MGBO

The hybrid PSOGBO algorithm leverages the strengths of both the PSO and M-MGBO algorithms to achieve a better performance. The PSO algorithm is the primary algorithm used for solving the optimization problem, while the M-MGBO algorithm is used to direct the population toward the optimal solution. Initially, at the start of the iteration, the PSO algorithm updates the velocity and position of each particle in the population. Next, the ratio of the cost (MSE) of the current solution to the global best solution cost is compared to a threshold. If this ratio exceeds the threshold, the particle is recorded and saved into a vector. This process is repeated for each particle in the population, and the newly generated vector is used as the initial population for the M-MGBO algorithm. The M-MGBO algorithm is then executed for a specified number of iterations, generating new members of the population that are compared to the original members to determine if the newly generated positions are improvements. Following this step, the global and personal best solutions are updated for the hybrid algorithm.

By combining the PSO algorithm’s efficiency in exploring the search space with the M-MGBO algorithm’s ability to guide the search towards the optimal solution, the hybrid approach achieves a better performance than either algorithm on its own. The algorithm flow chart is shown in Figure 8.



**Figure 8.** Flow chart hybrid PSO-MGBO method developed in this work.  $P_{best}$  is the personal best solution, and  $G_{Best}$  is the global best solution. Text in red indicates the related steps to the M-MGBO algorithm.

### 3.3.3. Adding the Local Escaping Operator to the PSO

Another variant of the PSO algorithm introduced in this research involves the incorporation of the local escaping operator (LCEO) into the standard PSO algorithm. The LCEO generates a superior particle that is guided by the global best solution ( $x_{best}$ ),  $x_{best2}$ , and  $x_{best3}$ , hence enhancing the diversity of the population. Similar to the MGBO and M-MGBO algorithms, the LCEO is used to overcome the problem of particles getting stuck in local optima by allowing the particle to escape from its current local optima and move toward the global optima.

After generating the LCEO solution  $X_{LCEO}$  (Algorithm 6), it is compared to the PSO-generated solution, the global best solution, and the personal best solution. If the performance of the  $X_{LCEO}$  solution in terms of MSE is better than the PSO-generated solution, the particle's current position is changed to the  $X_{LCEO}$  position, and the particle's velocity is computed again. Otherwise, the  $X_{LCEO}$  solution is discarded, and the particle's current position remains unchanged. This process is repeated until the optimal solution is found. Algorithm 7 provides the pseudocode of the algorithm applied for SC parameterization. The use of LCEO in the PSO algorithm has exhibited a promising performance in tackling optimization problems, particularly in scenarios where the search space is complex or where particles tend to get trapped in local optima.

---

**Algorithm 7** PSO-LCEO
 

---

**Require:**  $nP$ ,  $n_j$ ,  $LC$ ,  $MaxIt$ ,  $X_{min}$ ,  $X_{max}$ ,  $C1$ ,  $C2$ ,  $W_{max}$ ,  $W_{min}$ ,  $V_{max}$ , and  $V_{min}$

```

1 Initialize set of Random population,  $P = [K_v, R_1, C_1, R_2, C_2, R_3, C_3, R_L]_{np \times nj}$ .
2 Evaluate the MSE for the population initially;  $MSE = \frac{1}{n} \sum_{i=1}^n (P_i - \tilde{P}_i)^2$ 
3 Select  $[G_{best}, P_{best}]$  from population // Such that  $G_{best} = \min(Cost)$  over entire
   population, and  $P_i^{best} = \operatorname{argmin}(Cost^i)$  at the  $i^{\text{th}}$  iteration.
4 for  $j < MaxIt$  do
5     Evaluate  $W$ ;  $W = W_{max} - \left( (W_{max} - W_{min}) * \frac{j}{MaxIt} \right)$ 
6     while  $i \leq nP$  do
7          $X_{old} = P_i^j$  //  $X_{old} = [K_v(i), R_1(i), C_1(i), R_2(i), C_2(i), R_3(i), C_3(i), R_L(i))_j$ 
8         Update  $V_i^j$  using (28)
9          $V_i^j = \operatorname{limit}(V_i^j)$  // Check for upper and lower bounds; Algorithm 4
10        Update  $P_i^j$  using (29)
11        Let  $A = [P_i^j(C_1), P_i^j(C_2), P_i^j(C_3)]_j$  and  $B = [P_i^j(R_1), P_i^j(R_2), P_i^j(R_3)]_j$ 
12         $P_i^j = \operatorname{limit}(P_i^j)$  // Check for upper and lower bounds; Algorithm 4
13         $P_i^j = \operatorname{Check}(A, B, X_{old})$  // Checking for constraints; Algorithm 5
14        Evaluate  $Cost(P_i^j)$  using Equation in line 2
15        if  $\operatorname{rand} < LC$  then // Local escaping operator Algorithm 6
16            Compute  $X_{LCEO}$  and the objective function of  $X_{LCEO}$ 
17             $X_{LCEO}^i = \operatorname{limit}(X_{LE}^i, P_i^j)$  // Check for upper and lower bounds
18            Let  $C = [X_{LCEO}^i(C_1^{LCEO}), X_{LCEO}^i(C_2^{LCEO}), X_{LCEO}^i(C_3^{LCEO})]_j$  and  $D =$ 
19             $[X_{LCEO}^i(R_1^{LCEO}), X_{LCEO}^i(R_2^{LCEO}), X_{LCEO}^i(R_3^{LCEO})]_j$ 
20             $X_{LCEO}^i = \operatorname{Check}(C, D, P_i^j)$  // Checking for constraints
21            Evaluate cost function for  $X_{LCEO}^j$  using Equation in line 2.
22             $P_i^j = \operatorname{Compare}(P_i^j, X_{LE}^i, Cost(P_i^j), Cost(X_{LE}^i))$  // To update  $P_i^j$  if
23             $Cost(X_{LE}^i) < Cost(P_i^j)$ 
24        end if
25        Update  $[G_{best}, P_i^{best}]$  // By comparing  $Cost(P_i^j)$  to  $Cost(G_{best})$  and
26         $Cost(P_i^{best})$ 
27    end while
28 end for

```

---

#### 4. Results and Discussion

In this study, the Zubieta model parameters were determined for an SC bank comprising six capacitors in series, each with a rating of 2300 F, resulting in a total bank capacitance of 383.33 F. The estimation process involved using three different input signals: a step input with amplitudes of 10 A (Figure 4a) and 4 A (Figure 4b), along with a variable frequency input (Figure 4c), corresponding to profiles 1, 2, and 3 respectively. For each input, four distinct algorithms (M-MGBO, PSO, PSOLCEO, and PSO-MGBO) were executed in three separate runs. The Zubieta model, characterized by three branches, requires the estimation of a total of eight parameters, namely  $R_1$ ,  $R_2$ ,  $R_3$ ,  $C_1$ ,  $C_2$ ,  $C_3$ ,  $Kv$ , and  $R_L$ . Each of the algorithms (M-MGBO, PSO, PSOLCEO, and PSO-MGBO) requires initial guesses, which are formed using the analytical Zubieta model parameters and also using any initial guess selection rules the algorithms may have. These are detailed in the listings of each algorithm—for example, certain algorithms use the random initialization of the parameters' values in the given bounds. The bounds themselves are derived using the parameters' values derived using Zubieta's analytical method. In each iteration of the respective algorithm, the parameters are estimated following the respective algorithmic procedures as outlined in Section 3. These estimated parameters are then used to estimate the supercapacitor terminal voltage at the given value of current, following Zubieta's model equations. The mean square error between the measured and estimated supercapacitor terminal voltage, which is obtained using the estimated parameters, is used as the cost/fitness function to evaluate the fitness of the estimated parameters.

It is worth noting that the existence of a voltage-dependent capacitance in one of the Zubieta model's branches makes this a nonlinear model. It is this nonlinear model that is used to evaluate the fitness of the estimated model parameters. Moreover, the MGBO algorithm used incorporates both gradient and population-based techniques to determine the search direction using Newton's method. This search direction is based on a linear estimation of the gradient of the cost/fitness function. However, such approximation can make the algorithm prone to being stuck in a local optimum. To avoid this, the local escaping operator (LCEO)-based routine is used to augment the MGBO algorithm into the M-MGBO algorithm.

##### 4.1. Parameters Estimation Results

The evaluation of the algorithms' performance considered various criteria, including MSE and the computation time. Optimization took place on a personal computer equipped with an Intel(R) Xeon(R) E-2124 central processing unit operating at a frequency of 3.30 gigahertz and 16 gigabytes of random-access memory.

The cost function for the optimization problem was set to minimize the difference between the simulated SC voltage from the derived model and the actual measured voltage.

$$\text{Minimize } SSE = \sum_{t=1}^T [V_i(t) - V_m(t)]^2, \quad (31)$$

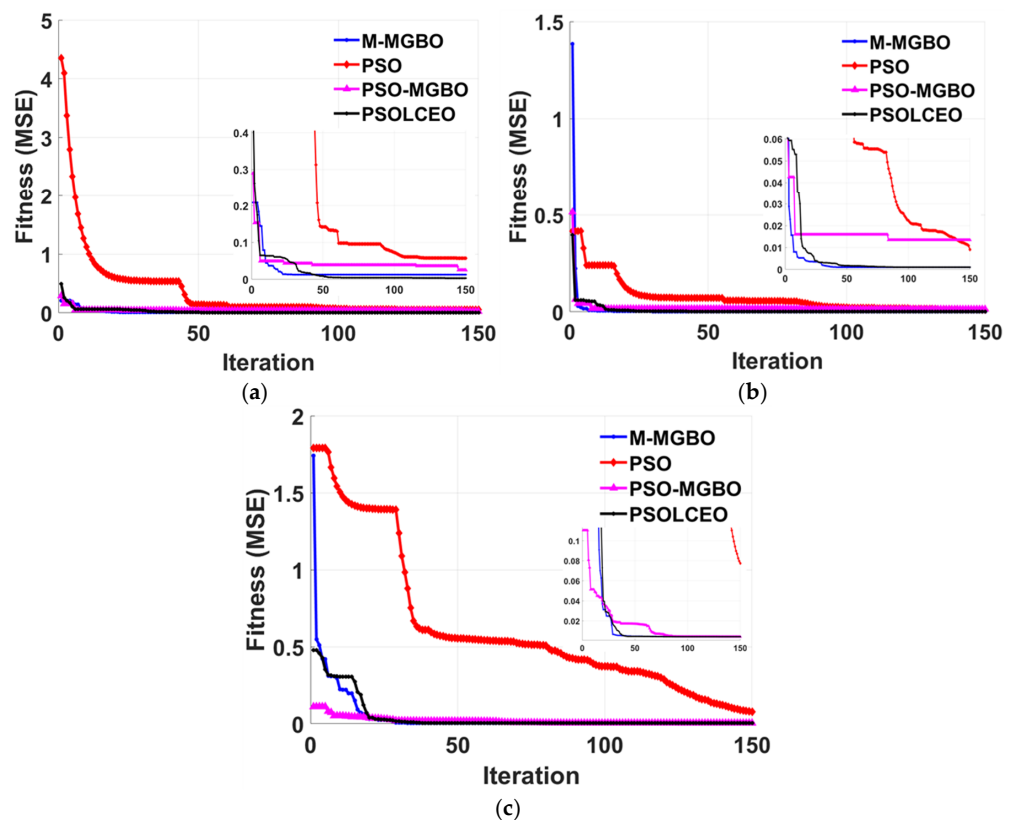
where  $V_i(t)$  is the capacitor terminal voltage given by Equation (7) obtained using estimated model parameters, and  $V_m(t)$  is the measured supercapacitor voltage.

The M-MGBO algorithm demonstrated superior performance compared with the other algorithms. In the case of Profile 1 (Table 3), M-MGBO and PSO-LCEO exhibited MSE values of 0.003927 and 0.003493, respectively. Notably, M-MGBO boasted the shortest computational time at 20,141.028 s, outperforming PSO-LCEO (26,782.93 s) and the hybrid PSO-MGBO (77,735.494 s). Additionally, parameters obtained by the M-MGBO algorithm were closer to analytically derived parameters (Table 2). For instance, the first branch capacitor  $C_1$  had a value of 0.7723 F with M-MGBO compared with 14.607195 F with PSO-LCEO.

**Table 3.** Experimental results for the parameters estimated using current Profile 1 with 150 iterations (T1, T2, and T3 are the respective branches' time constants).

Parameter	M-MGBO	PSO	PSO-LCEO	PSO-MGBO
$K_v \left( \frac{F}{V} \right)$	38.45929984	14.07398	33.22398	33.266903
$R_1 (\Omega)$	0.013163046	0.127028	0.014034	0.04586
$C_1 (F)$	0.77232877	48.13062	14.60719	0.95056
$R_2 (\Omega)$	0.487789546	5.190033	0.4396	1.61732
$C_2 (F)$	224.0938314	154.1119	168.6782	175.7074
$R_3 (\Omega)$	36.734606504	21.07912	4.807739	42.54308
$C_3 (F)$	394.158958	331.8499	214.2253	243.417
$R_l (\Omega)$	25.70462833	68.13273	62.47676	37.94408
MSE	0.003927391	0.057314	0.003494	0.025034
Elapsed Time	20141.028	16564.040	26782.930	77735.494
T1	0.010166	6.113936	0.204997	0.043593
T2	109.3106	799.8458	74.15094	284.1751
T3	14479.27	6995.104	1029.939	10355.71
Average MSE	0.006813	0.07581693	0.008789473	0.03842393

The convergence curve plays a vital role in assessing optimization algorithms, offering insights into the speed and efficiency of their convergence to the optimal solution. Notably, the convergence curve for the M-MGBO algorithm stands out for its swift and stable convergence with minimal oscillation. While the PSO-LCEO algorithm also showed stable convergence, it did so at a slower pace compared with the M-MGBO algorithm. For Profile 1, Figure 9a illustrates the convergence curve of the best run for all four algorithms, showcasing the M-MGBO algorithm's rapid convergence. Simultaneously, Figure 10a presents the voltage response for the same input.



**Figure 9.** (a) Convergence curve for Profile 1, (b) Convergence curve for Profile 2, and (c) Convergence curve for Profile 3.

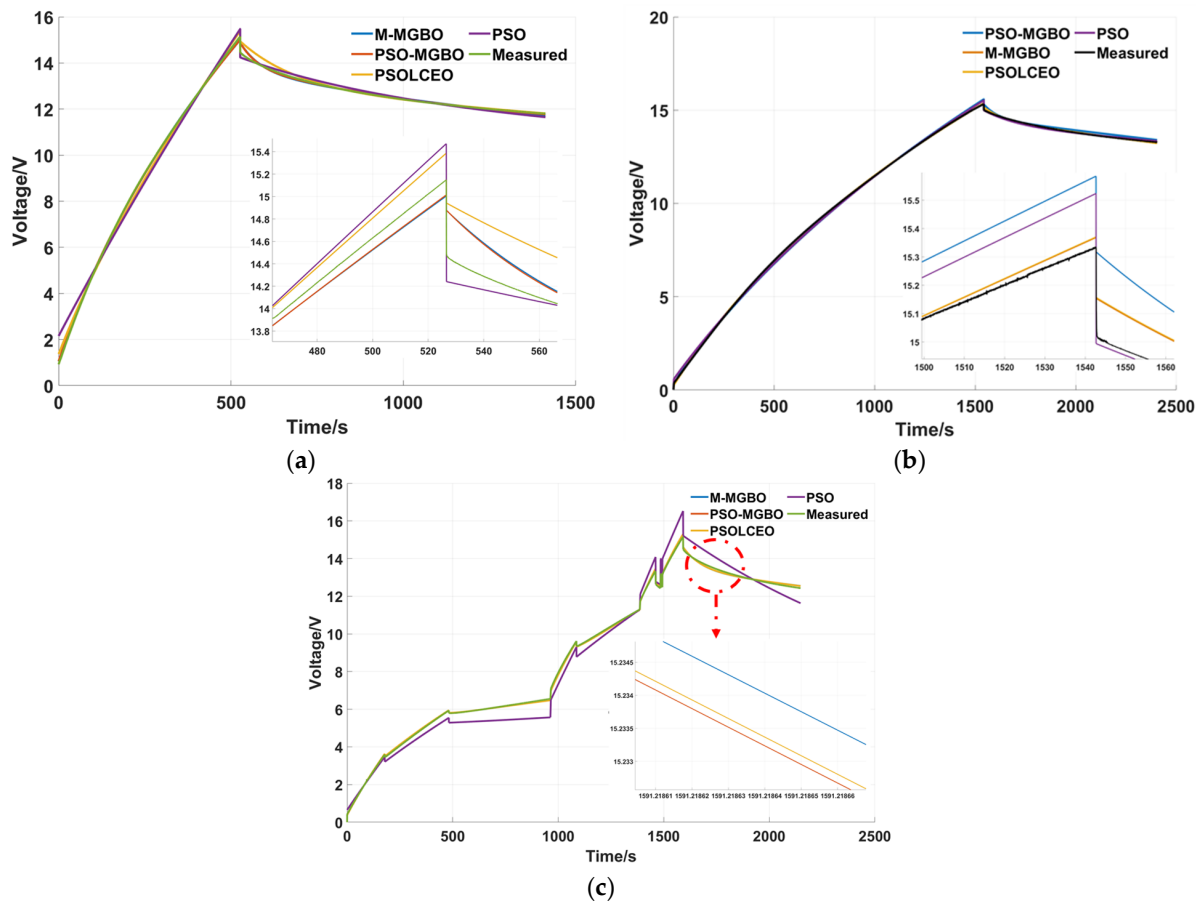


Figure 10. (a) Voltage response for Profile 1, (b) Voltage response for Profile 2, and (c) Voltage response for Profile 3.

Table 4 demonstrates the performance of the four optimization algorithms in estimating the parameters of a Zubieta supercapacitor model using Profile 2 as input. The M-MGBO algorithm exhibited the lowest MSE of 0.001001 and a significantly shortened computational time of 31,320.07 s. In contrast, the PSO-LCEO algorithm yielded a very similar MSE of 0.001017 compared with the M-MGBO algorithm. However, it exhibited a longer execution time of 36,297.98 s.

Table 4. Experimental results for the parameters estimated using current Profile 2 with 150 iterations (T1, T2, and T3 are the respective branches' time constants).

Parameter	M-MGBO	PSO	PSO-LCEO	PSO-MGBO
$K_v \left( \frac{F}{V} \right)$	37.2805859	27.8131	36.4359	36.86813
$R_1 (\Omega)$	0.05835297	0.07556	0.056184	0.149894
$C_1 (F)$	0.03307588	17.957	0.627231	0.105358
$R_2 (\Omega)$	0.69575519	0.5943	0.701702	5.188335
$C_2 (F)$	228.197898	163.674	226.8816	198.3325
$R_3 (\Omega)$	24.4703793	31.6255	18.62833	37.35831
$C_3 (F)$	394.767419	330.281	273.4351	251.036
$R_l (\Omega)$	52.358648	56.0493	94.7854	63.34033
MSE	0.0010013	0.01342	0.001017	0.009034
Elapsed Time	31320.07	109260	36297.98	22324.22
T1	0.00193	1.356831	0.03524	0.015793
T2	158.7699	97.27146	159.2033	1029.015
T3	9660.108	10445.3	5093.639	9378.281
Average MSE	0.001001	0.103347	0.0031693	0.107982

The hybrid PSO-MGBO and standard PSO algorithms exhibited higher MSE values of 0.01342 and 0.009034, respectively. In terms of average MSE, the M-MGBO algorithm outperformed others with a value of 0.001001, followed by the PSOL-CEO algorithm at 0.003169. The hybrid PSO-MGBO and regular PSO algorithms had higher average MSE values at 0.103347 and 0.107982, respectively. It is noteworthy that the LCEO operator, which is employed in the PSO-LCEO algorithm, serves as a method for escaping local optima during the optimization process. Despite this feature, the PSO-LCEO algorithm could not provide superior parameter estimates compared with the M-MGBO algorithm in this study. For the convergence curve of the best run for the four algorithms, refer to Figure 9b, while Figure 10b presents the corresponding voltage response.

The parameters obtained using Profile 3 input are presented in Table 5. The M-MGBO algorithm exhibited an MSE of 0.0042315, while the PSO-LCEO showed a comparable performance with an MSE of 0.004231. The hybrid PSO-MGBO had a similar MSE of 0.004231 but still outperformed the PSO-LCEO in terms of computational time, completing the task in only 32,206 s compared with 39,252 s. Notably, the M-MGBO algorithm showcased the best performance in terms of simulation time, with a runtime of 28,622 s.

**Table 5.** Experimental results for the parameters estimated using a variable frequency current profile with 150 iterations (T1, T2, and T3 are the respective branches' time constants).

Parameter	M-MGBO	PSO	PSO-LCEO	PSO-MGBO
$K_v \left( \frac{F}{V} \right)$	51.74533692	125.415	51.72809	51.74644
$R_1 (\Omega)$	0.080265842	0.1346	0.080451	0.015515
$C_1 (F)$	0.012783609	71.44	0.002166	0.572341
$R_2 (\Omega)$	0.572682701	8.90377	0.571799	0.080359
$C_2 (F)$	223.6858074	186.193	223.5703	223.6356
$R_3 (\Omega)$	49.99947713	24.526	50	49.99997
$C_3 (F)$	399.9692153	292.147	400	400
$R_l (\Omega)$	20.90658189	31.4604	20.86642	20.90162
MSE	0.004231544	0.30036	0.004231	0.004231
Elapsed Time	28622.51195	19408.00527	39252.59362	32206.02127
T1	0.001026	9.615824	0.000174	0.00888
T2	128.101	1657.82	127.8373	17.97113
T3	19998.25	7165.197	20000	19999.99
Average MSE	0.004233505	0.226921781	0.010902	0.004257128

It can be noted from the results that the M-MGBO and PSO-LCEO algorithms showed the best performance in terms of MSE, while the PSO-MGBO algorithm had a good balance between performance and simulation time. The PSO algorithm had the lowest computational time, but it had a significantly worse performance in terms of MSE. The convergence curve for the best run of the four algorithms is presented in Figure 9c, and the voltage response is presented in Figure 10c.

#### 4.2. M-MGBO Results Validation

The results of 30 iterations of the M-MGBO method used to estimate the parameters of the Zubieta model of supercapacitors using a variable frequency input are presented in Table 6. The primary objective of this analysis is to assess the algorithm's performance in terms of the consistency of parameter estimations, the accuracy of results pertaining to resistance and capacitance values, and any other noteworthy observations obtained from the table.



**Table 6.** Experimental results for the parameters using M-MGBO and variable frequency with 30 runs (T1, T2, and T3 are the respective branches' time constants).

Parameter	Standard Deviation	Average
$K_v \left( \frac{F}{V} \right)$	0.346049	51.95607
$R_1 (\Omega)$	0.000967	0.079637
$C_1 (F)$	0.005086	0.01552
$R_2 (\Omega)$	0.02133	0.555367
$C_2 (F)$	0.129872	223.7431
$R_3 (\Omega)$	0.070908	49.96596
$C_3 (F)$	0.912619	399.2558
$R_l (\Omega)$	0.405767	21.18428
T1	0.000393	0.001233
T2	4.766871	124.2593
T3	72.75695	19949.26

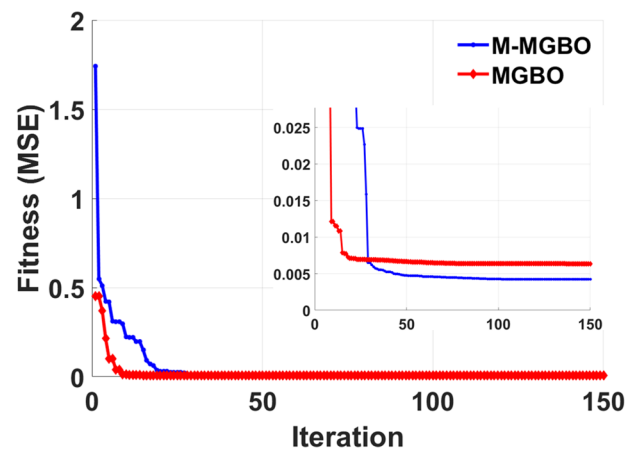
It is also important to consider the parameters' standard deviation while analyzing the algorithm's performance, as it provides a measure of the algorithm's consistency. The standard deviation is a statistical measure used to assess the variability in parameter estimations across the 30 runs. Lower values of standard deviation indicate a higher level of consistency in the estimations. It is worth mentioning that certain parameters, including  $R_1 (\Omega)$  and  $R_2 (\Omega)$ , exhibit remarkably low standard deviation values, specifically 0.0009 and 0.02, respectively. The observed low values of standard deviation suggest that the M-MGBO algorithm exhibits a consistent pattern of convergence, yielding parameter estimations that are highly similar across every iteration. The level of consistency demonstrated in this context is of the utmost importance for practical applications, as it ensures the reliability and predictability of outcomes.

#### 4.3. MGBO and M-MGBO Comparison

To verify the reliability and assess the effectiveness of the modifications applied to the original MGBO algorithm, three separate runs were executed using the unaltered MGBO algorithm but with varying frequency inputs. The MGBO algorithm produced an average MSE of 0.0064, with a standard deviation of  $5.7735 \times 10^{-7}$  over the course of these three runs. Table 7 provides a comparative analysis between the M-MGBO and MGBO. Additionally, Figure 11 illustrates the convergence curve for the best-performing runs.

**Table 7.** Comparison of M-MGBO and M-MGBO best run 150 iterations (T1, T2, and T3 are the respective branches' time constants).

Parameter	MGBO	M-MGBO
$K_v \left( \frac{F}{V} \right)$	53.00867	51.74533692
$R_1 (\Omega)$	0.229319	0.080265842
$C_1 (F)$	0.0488278	0.012783609
$R_2 (\Omega)$	0.872346	0.572682701
$C_2 (F)$	223.6373	223.6858074
$R_3 (\Omega)$	50	49.99947713
$C_3 (F)$	399.9993	399.9692153
$R_l (\Omega)$	20.90988	20.90658189
T1	127.9978	17.9543
T2	19999.97	19998.2516
T3	53.00867	51.74533692
MSE	0.0063571	0.004231544



**Figure 11.** Best run convergence curve comparison between M-GBO and M-MGBO for variable frequency input.

For  $Kv$  the MGBO yielded a value of 53.0087 F/V, while the M-MGBO resulted in 51.7453 F/V. Despite the standard M-MGBO exhibiting a greater MSE when compared to the M-MGBO, it demonstrated a lower standard deviation for  $Kv$ . Specifically, the standard deviation for  $Kv$  was 0.0702 for the standard MGBO and 0.2311 for M-MGBO. This pattern is consistent for both  $R3$  and  $RL$ .

#### 4.4. Parameters Testing

As demonstrated earlier, the M-MGBO method provides a more precise estimation of parameters concerning accuracy and MSE. To conduct additional testing on parameters from the four distinct methods, the optimal parameters acquired from the three runs (as presented in Tables 4–6) undergo further evaluation by interchanging and applying them to the remaining inputs. For example, parameters obtained using Profile 1 input are tested using Profile 2 and Profile 3 as inputs, and their mean squared error (MSE) is recorded.

Table 8 presents a scenario where parameters, initially estimated via Profile 1, were subsequently evaluated with Profile 2 and Profile 3 inputs. In the experiment with Profile 2, it is evident that the parameters of the M-MGBO algorithm yield the lowest MSE value of 1.4291. Furthermore, the PSO, PSO-MGBO, and PSO-LCEO had similar MSE values. Specifically, the PSO-LCEO algorithm demonstrated the second lowest MSE of 1.6312, followed by the PSO-MGBO method, with an MSE of 1.6396, and the PSO algorithm, with an MSE of 1.6442. When considering the use of parameters with Profile 3, it was seen that the M-MGBO algorithm exhibited the lowest MSE of 0.1195. Subsequently, the PSO-LCEO program showed an MSE of 0.1469, while the PSOGBO algorithm yielded an MSE of 0.1482. The PSO method demonstrated the most significant MSE value, measuring 0.4614.

**Table 8.** MSE for Profile 1 parameters tested with Profiles 2 and 3 as inputs.

Algorithm	Profile 2 (4A Input)	Profile 3 (Variable Frequency Input)
MGBO	----	0.1489
PSO-GBO	1.6396	0.1482
M-MGBO	1.4291	0.1195
PSO-LCEO	1.6312	0.1469
PSO	1.6442	0.4614

Table 9 presents the results of the parameter testing conducted using parameters estimated using Profile 3 but tested with Profile 1 and Profile 2. The M-MGBO parameters yielded the lowest MSE in all tests. Specifically, the MSE was found to be 0.7448 for the Profile 2 input and 0.1754 for the Profile 1 input. The parameters of the PSO-LCEO

algorithm exhibited the second-lowest MSE for both tests. Specifically, the MSE was found to be 0.7464 for the 4 A step input and 0.2001 for the 10 A step input.

**Table 9.** MSE for Profile 3 parameters tested with Profiles 1 and 2 as inputs.

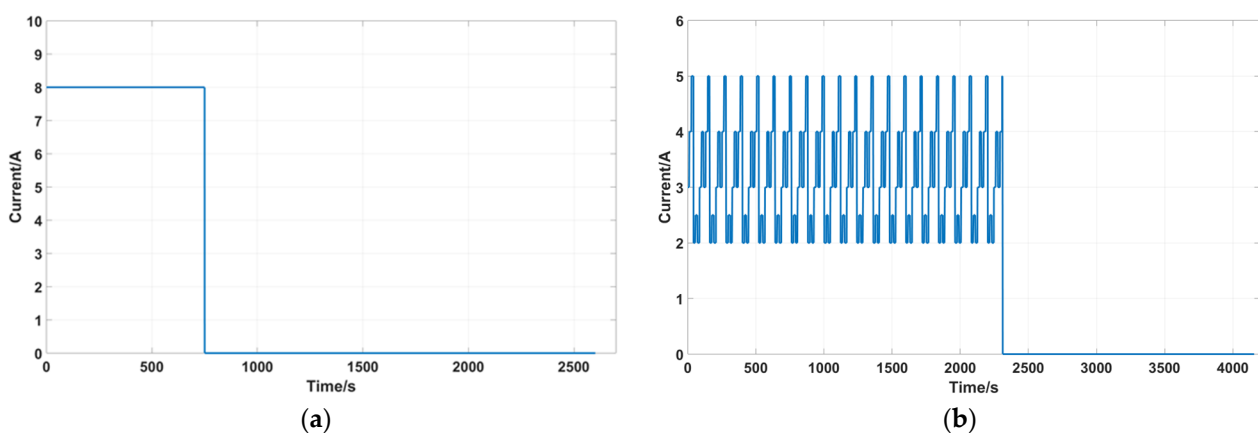
Algorithm	Profile 2 (4 A Input)	Profile 1 (10 A Input)
MGBO	----	----
PSO-MGBO	0.8144	0.5252
M-MGBO	0.7448	0.1754
PSOL-CEO	0.7464	0.2001
PSO	0.8645	0.5794

The PSO-MGBO algorithm parameters have been determined to be the third-best performing algorithm, whereas the PSO algorithm exhibited the greatest mean MSE for both test cases. In a similar manner, while employing parameters estimated using Profile 2 tested with Profile 1 as input (Table 10), the same pattern emerged. The M-MGBO algorithm parameters yielded the lowest MSE value of 1.4149, followed by the PSO-LCEO and the PSO-MGBO, with MSE values of 1.5438 and 1.5349, respectively. In the instance of the Profile 3 input, the PSOL-CEO parameters yielded the lowest MSE of 0.9877, while the M-MGBO resulted in a slightly higher MSE of 0.9902. Furthermore, the particle swarm optimization (PSO) method yielded the highest MSE for both test scenarios.

**Table 10.** M-MGBO variable frequency 30 runs parameters standard deviation and average.

Algorithm	Profile 1 (10 A Input)	Profile 3 (Variable Frequency Input)
MGBO	----	1.2619
PSO-MGBO	1.5349	0.9896
M-MGBO	1.4149	0.9902
PSO-LCEO	1.5438	0.9877
PSO	1.6930	1.2362

To perform a more exhaustive evaluation of the algorithm's parameters, two supplementary tests were conducted. These tests involved using a step input of 8 A (See Figure 12a) and a controlled current sequence as inputs. The optimal run parameters derived from prior experiments were used in both tests. Additionally, the performance is also assessed by comparing it to the analytical technique in order to validate the suitability of the parameters derived from the four algorithms.



**Figure 12.** (a) 8 A step input, (b) Controlled current sequence.

As evident from Table 11, concerning the Profile 2 parameters, the M-MGBO algorithm demonstrates the lowest MSE at 0.9619, followed by the PSOLCEO with the second lowest

MSE of 0.9793, and subsequently, the PSO-MGBO algorithm. In the case of the Profile 1 parameters, the PSO-LCEO showcased the lowest MSE of 0.1418, followed by the PSO-MGBO algorithm, while the M-MGBO ranked third, with an MSE of 0.2699. Profile 3 parameters exhibited a closely aligned response among M-MGBO, PSO-LCEO, and PSO-MGBO, displaying minimal deviation in MSE values. Conversely, the PSO algorithm, across various inputs, performed the least favorably, indicating higher MSE values and implying lower accuracy and reliability in the estimated parameter.

**Table 11.** Terminal voltage MSE for test using 8A profile, with parameters estimated using profiles 1, 2, and 3 and the analytical approach parameters.

Algorithm	Profile 2 (4 A Parameters)	Profile 1 (10 A Parameters)	Profile 3 (Variable Frequency Parameters)
PSO-MGBO	0.9848	0.1617	0.0942
M-MGBO	0.9619	0.2699	0.0943
PSO-LCEO	0.9793	0.1418	0.0945
PSO	1.1325	0.2824	0.3063
Analytical		7.7745	

The controlled current sequence profile used for the following tests is illustrated in Figure 12b. According to the data presented in Table 12, it can be observed that the M-MGBO algorithm exhibited superior performance compared with the other algorithms and the analytical approach in terms of Profile 2 parameters. Specifically, the M-MGBO algorithm achieved an MSE value of 1.5037.

**Table 12.** Terminal voltage MSE for test using controlled current sequence profile, with parameters estimated using profiles 1, 2, and 3 and the analytical approach parameters.

Algorithm	Profile 2 (4 A Parameters)	Profile 1 (10 A Parameters)	Profile 3 (Variable Frequency Parameters)
PSO-MGBO	2.1681	0.5573	0.0571
M-MGBO	1.5037	0.0562	0.0483
PSO-LCEO	1.5778	0.2005	0.05765
PSO	2.1251	0.3547	0.0570
Analytical		7.364942	

In the case of the Profile 1 parameters, it is evident that the M-MGBO algorithm exhibited superior performance compared with the other algorithms. The M-MGBO algorithm achieved an MSE of 0.0562, while the PSO-LCEO algorithm achieved an MSE of 0.2005, and the PSO-MGBO algorithm achieved an MSE of 0.5573. Nevertheless, the PSO-MGBO algorithm had the lowest performance in this particular test, as the PSO algorithm outperformed it, with an MSE value of 0.3547. Moreover, using Profile 3 parameters with the controlled current sequence profile, it was observed that the M-MGBO algorithm demonstrated superior performance in comparison to the other algorithms, achieving an MSE of 0.0483. On the other hand, the PSO, PSO-LCEO, and PSO-MGBO algorithms demonstrated a similar performance, with minimal deviation in MSE values.

## 5. Conclusions

The importance of energy storage for grid stability and power backup in both conventional and renewable systems is evident. Particularly, supercapacitors stand out for their excellence in high-power density applications, such as regenerative braking in electric vehicles. This research concentrates on the parameterization of the Zubieta model tailored for supercapacitors, employing a hybrid metaheuristic gradient-based optimization (MGBO) strategy. The Zubieta model, featuring three RC branches and a self-discharge branch, necessitates the identification of seven parameters. The investigation systematically

compared the performance of the modified MGBO (M-MGBO) against particle swarm optimization (PSO) and two variations—one integrating PSO with M-MGBO and the other introducing a local escaping operator (LCEO). The evaluation encompasses key metrics like convergence rate, accuracy, and processing/computational time. The results indicate that both hybrid PSO-MGBO and PSO-LCEO surpass conventional PSO, exhibiting enhancements of 51% and 94%, respectively. Remarkably, these variants proved to be comparable to M-MGBO. These methodologies effectively contribute to the accurate estimation of Zubieta model parameters. The study underscores the considerable potential of hybrid optimization strategies in advancing the precision and effectiveness of supercapacitor model parameterization.

The findings showed M-MGBO's adaptability to diverse operational contexts, with specific examples of a computational time of 31,320.07 s under a 4 A step input. Delving into the consistency and reliability of M-MGBO, the study scrutinized standard deviation values across 30 runs, revealing a remarkably stable pattern. This consistency is pivotal for practical applications, ensuring predictability and reliability in outcomes. The meticulous examination of the algorithm's performance underscored its reliability, with a computational time of approximately 28,610 s across 30 runs, adhering to the expected order of magnitude in resistance and capacitance values with respect to the analytical approach. As reported in the literature, when using metaheuristic algorithms the search for the optimal solution can become caught in local optima in most nonlinear systems with more than one minimum. Moreover, they do not always guarantee the global optimum solution and only provide nearly optimal solutions. This paper attempts to overcome these challenges by proposing some novel updates to the metaheuristic algorithms, and further improvements are left for future efforts.

**Author Contributions:** Conceptualization, A.Y., R.D. and S.M.; methodology, A.Y., R.D. and S.M.; software, A.Y.; validation, A.Y.; formal analysis, A.Y., R.D. and S.M.; investigation, A.Y., R.D. and S.M.; resources, A.Y., R.D. and S.M.; data curation, A.Y.; writing—original draft preparation, A.Y.; writing—review and editing, A.Y., R.D. and S.M.; visualization, A.Y.; supervision, R.D. and S.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** The work in this paper was supported, in part, by the Open Access Program from the American University of Sharjah.

**Data Availability Statement:** All data required can be generated using the equations/algorithms presented in this work. Experimental data requires experimentation with the appropriate hardware apparatus as shown in the paper.

**Acknowledgments:** This paper represents the opinions of the authors and does not mean to represent the position or opinions of the American University of Sharjah.

**Conflicts of Interest:** Authors declare no conflicts of interest.

## References

1. Sayed, E.T.; Olabi, A.G.; Alami, A.H.; Radwan, A.; Mdallal, A.; Rezk, A.; Abdelkareem, M.A. Renewable Energy and Energy Storage Systems. *Energies* **2023**, *16*, 1415. [[CrossRef](#)]
2. Olabi, A.; Onumaegbu, C.; Wilberforce, T.; Ramadan, M.; Abdelkareem, M.A.; Alami, A.H.A. Critical review of energy storage systems. *Energy* **2021**, *214*, 118987. [[CrossRef](#)]
3. Salameh, T.; Abdelkareem, M.A.; Olabi, A.G.; Sayed, E.T.; Al-Chaderchi, M.; Rezk, H. Integrated standalone hybrid solar PV, fuel cell and diesel generator power system for battery or supercapacitor storage systems in Khorfakkan, United Arab Emirates. *Int. J. Hydrogen Energy* **2021**, *46*, 6014–6027. [[CrossRef](#)]
4. Rezk, H.; Nassef, A.M.; Abdelkareem, M.A.; Alami, A.H.; Fathy, A. Comparison among various energy management strategies for reducing hydrogen consumption in a hybrid fuel cell/supercapacitor/battery system. *Int. J. Hydrogen Energy* **2021**, *46*, 6110–6126. [[CrossRef](#)]
5. Adib, A.; Dhaouadi, R. Modeling and analysis of a regenerative braking system with a battery-supercapacitor energy storage. In Proceedings of the 2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Sharjah, United Arab Emirates, 4–6 April 2017. [[CrossRef](#)]
6. Zhang, J.; Gu, M.; Chen, X. Supercapacitors for renewable energy applications: A review. *Micro Nano Eng.* **2023**, *21*, 100229. [[CrossRef](#)]

7. Naseri, F.; Karimi, S.; Farjah, E.; Schaltz, E. Supercapacitor management system: A comprehensive review of modeling, estimation, balancing, and protection techniques. *Renew. Sustain. Energy Rev.* **2022**, *155*, 111913. [[CrossRef](#)]
8. Haris, M.; Hasan, M.N.; Qin, S. State of health prediction of supercapacitors using multi-trend learning of NARX neural network. *Mater. Today Sustain.* **2022**, *20*, 100201. [[CrossRef](#)]
9. Liu, C.; Li, Q.; Wang, K. State-of-charge estimation and remaining useful life prediction of supercapacitors. *Renew. Sustain. Energy Rev.* **2021**, *150*, 111408. [[CrossRef](#)]
10. Mehta, U.; Prasad, R.; Kothari, K. Various analytical models for supercapacitors: A mathematical study. *Resour. Technol.* **2020**, *1*, 1–15. [[CrossRef](#)]
11. Mukhopadhyay, S.; Dhaouadi, R.; Takrouiri, M.; Dogga, R. Supercapacitor Characterization Using Universal Adaptive Stabilization and Optimization. *IEEE Open J. Ind. Electron. Soc.* **2020**, *1*, 166–183. [[CrossRef](#)]
12. Chen, Y.; Yan, R.; Yu, C.; Zhao, C.; Huang, X.; Wei, L. Investigation on Characteristic Parameters Identification and Evolution of Supercapacitor Energy Storage System From Sparse and Fragmented Monitoring Data. *IEEE Access* **2023**, *11*, 56983–56993. [[CrossRef](#)]
13. Zhang, L.; Hu, X.; Wang, Z.; Sun, F.; Dorrell, D.G. A review of supercapacitor modeling, estimation, and applications: A control/management perspective. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1868–1878. [[CrossRef](#)]
14. Hetzel, M.; Ocampo, D.D.; De Carne, G.; Hiller, M. Supercapacitor Modeling and Parameter Identification of a 400 kW Grid-Connected Supercapacitor Energy Storage System using the Inherent Impedance Spectroscopy Capability of its DC/DC Converter. In Proceedings of the 2023 IEEE Energy Conversion Congress and Exposition (ECCE), Nashville, TN, USA, 29 October 29–2 November 2023; pp. 459–465.
15. Majchrzycki, W.; Jankowska, E.; Baraniak, M.; Handzlik, P.; Samborski, R. Electrochemical Impedance Spectroscopy and Determination of the Internal Resistance as a Way to Estimate Lead-Acid Batteries Condition. *Batteries* **2018**, *4*, 70. [[CrossRef](#)]
16. Barcellona, S.; Ciccarelli, F.; Iannuzzi, D.; Piegari, L. Modeling and parameter identification of lithium-ion capacitor modules. *IEEE Trans. Sustain. Energy* **2014**, *5*, 785–794. [[CrossRef](#)]
17. Chatterjee, K.; Agnihotri, P.K.; Basu, S.; Gupta, N. Quantification of Pore Accessibility in Mesoporous Supercapacitor Electrode Using Cyclic Voltammetry. *IEEE Trans. Instrum. Meas.* **2023**, *73*, 3341125. [[CrossRef](#)]
18. Zucca, M.; Hassanzadeh, M.; Conti, O.; Pogliano, U. Accurate Parameters Identification of a Supercapacitor Three-Branch Model. *IEEE Access* **2023**, *11*, 122387–122398. [[CrossRef](#)]
19. Marín-Coca, S.; Ostadrahimi, A.; Bifaretti, S.; Roibás-Millán, E.; Pindado, S. New Parameter Identification Method for Supercapacitor Model. *IEEE Access* **2023**, *11*, 21771–21782. [[CrossRef](#)]
20. Zubieta, L.; Bonert, R. Characterization of double-layer capacitors for power electronics applications. *IEEE Trans. Ind. Appl.* **2000**, *36*, 199–205. [[CrossRef](#)]
21. Krishnan, G.; Das, S.; Agarwal, V. An Online Identification Algorithm to Determine the Parameters of the Fractional-Order Model of a Supercapacitor. *IEEE Trans. Ind. Appl.* **2020**, *56*, 763–770. [[CrossRef](#)]
22. Wang, J.; Zhang, L.; Mao, J.; Zhou, J.; Xu, D. Fractional Order Equivalent Circuit Model and SOC Estimation of Supercapacitors for Use in HESS. *IEEE Access* **2019**, *7*, 52565–52572. [[CrossRef](#)]
23. Reichbach, N.; Kuperman, A. Recursive-Least-Squares-Based Real-Time Estimation of Supercapacitor Parameters. *IEEE Trans. Energy Convers.* **2016**, *31*, 810–812. [[CrossRef](#)]
24. Xu, D.; Zhang, L.; Wang, B.; Ma, G. A novel equivalent-circuit model and parameter identification method for supercapacitor performance. *Energy Procedia* **2018**, *145*, 133–138. [[CrossRef](#)]
25. Xu, D.; Zhang, L.; Wang, B.; Ma, G. Modeling of Supercapacitor Behavior with an Improved Two-Branch Equivalent Circuit. *IEEE Access* **2019**, *7*, 26379–26390. [[CrossRef](#)]
26. Wei, T.; Qi, X.; Qi, Z. An improved ultracapacitor equivalent circuit model for the design of energy storage power systems. In Proceedings of the 2007 International Conference on Electrical Machines and Systems, Seoul, Republic of Korea, 8–11 October 2007; pp. 69–73. [[CrossRef](#)]
27. Pucci, M.; Vitale, G.; Cirrincione, G.; Cirrincione, M. Parameter identification of a Double-Layer-Capacitor 2-branch model by a least-squares method. In Proceedings of the IECON 2013—39th Annual Conference of the IEEE Industrial Electronics Society, Vienna, Austria, 10–13 November 2013; pp. 6770–6776. [[CrossRef](#)]
28. Faranda, R. A new parameters identification procedure for simplified double layer capacitor two-branch model. *Electr. Power Syst. Res.* **2010**, *80*, 363–371. [[CrossRef](#)]
29. Zhang, L.; Wang, Z.; Sun, F.; Dorrell, D.G. Online Parameter Identification of Ultracapacitor Models Using the Extended Kalman Filter. *Energies* **2014**, *7*, 3204–3217. [[CrossRef](#)]
30. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [[CrossRef](#)]
31. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
32. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
33. Mirjalili, S. The Ant Lion Optimizer. *Adv. Eng. Softw.* **2015**, *83*, 80–98. [[CrossRef](#)]
34. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, CA, USA, 1992.

35. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2020**, *32*, 9383–9425. [[CrossRef](#)]
36. Nassef, A.M.; Fathy, A.; Rezk, H.; Yousri, D. Optimal parameter identification of supercapacitor model using bald eagle search optimization algorithm. *J. Energy Storage* **2022**, *50*, 104603. [[CrossRef](#)]
37. Fathy, A.; Rezk, H. Robust electrical parameter extraction methodology based on Interior Search Optimization Algorithm applied to supercapacitor. *ISA Trans.* **2020**, *105*, 86–97. [[CrossRef](#)]
38. Gandomi, A.H. Interior search algorithm (ISA): A novel approach for global optimization. *ISA Trans.* **2014**, *53*, 1168–1183. [[CrossRef](#)]
39. Sun, G.; Liu, Y.; Chai, R.; Mei, F.; Zhang, Y. Online model parameter identification for supercapacitor based on weighting bat algorithm. *AEU—Int. J. Electron. Commun.* **2018**, *87*, 113–118. [[CrossRef](#)]
40. Mostafaoui, M.; Benyoucef, D. Electrical model parameters identification of radiofrequency discharge in argon through 1D3V/PIC-MC model. *Plasma Sci. Technol.* **2018**, *20*, 095401. [[CrossRef](#)]
41. Zhao, Y.; Xie, W.; Fang, Z.; Liu, S. A Parameters Identification Method of the Equivalent Circuit Model of the Supercapacitor Cell Module Based on Segmentation Optimization. *IEEE Access* **2020**, *8*, 92895–92906. [[CrossRef](#)]
42. Goh, C.-T.; Cruden, A. Bivariate quadratic method in quantifying the differential capacitance and energy capacity of supercapacitors under high current operation. *J. Power Sources* **2014**, *265*, 291–298. [[CrossRef](#)]
43. Zhao, Y.; Wei, L. Structure and Parameter Identification of Supercapacitors Based on Particle Swarm Optimization. *Zhongguo Dianji Gongcheng Xuebao* **2012**, *32*, 155–161.
44. Raouti, D.; Flazi, S.; Benyoucef, D. Modeling and Identification of Electrical Parameters of Positive DC Point-to-Plane Corona Discharge in Dry Air Using RLS Method. *IEEE Trans. Plasma Sci.* **2016**, *44*, 1144–1149. [[CrossRef](#)]
45. Mirjalili, S. SCA: A Sine Cosine Algorithm for solving optimization problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [[CrossRef](#)]
46. Abbassi, R.; Saidi, S.; Abbassi, A.; Jerbi, H.; Kchaou, M.; Alhasnawi, B.N. Accurate Key Parameters Estimation of PEMFCs' Models Based on Dandelion Optimization Algorithm. *Mathematics* **2023**, *11*, 1298. [[CrossRef](#)]
47. Abbassi, R.; Saidi, S.; Urooj, S.; Alhasnawi, B.N.; Alawad, M.A.; Premkumar, M. An Accurate Metaheuristic Mountain Gazelle Optimizer for Parameter Estimation of Single- and Double-Diode Photovoltaic Cell Models. *Mathematics* **2023**, *11*, 4565. [[CrossRef](#)]
48. Ahmadianfar, I.; Bozorg-Haddad, O.; Chu, X. Gradient-based optimizer: A new metaheuristic optimization algorithm. *Inf. Sci.* **2020**, *540*, 131–159. [[CrossRef](#)]
49. Ahmadianfar, I.; Gong, W.; Heidari, A.A.; Golilarz, N.A.; Samadi-Kouchehsaraee, A.; Chen, H. Gradient-based optimization with ranking mechanisms for parameter identification of photovoltaic systems. *Energy Rep.* **2021**, *7*, 3979–3997. [[CrossRef](#)]
50. Shami, T.M.; El-Saleh, A.A.; Alswaitti, M.; Al-Tashi, Q.; Summakieh, M.A.; Mirjalili, S. Particle Swarm Optimization: A Comprehensive Survey. *IEEE Access* **2022**, *10*, 10031–10061. [[CrossRef](#)]
51. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [[CrossRef](#)]
52. Hafez, I.; Dhaouadi, R. Identification of Mechanical Parameters in Flexible Drive Systems Using Hybrid Particle Swarm Optimization Based on the Quasi-Newton Method. *Algorithms* **2023**, *16*, 371. [[CrossRef](#)]
53. Usman, H.M.; Mukhopadhyay, S.; Rehman, H. Universal Adaptive Stabilizer Based Optimization for Li-Ion Battery Model Parameters Estimation: An Experimental Study. *IEEE Access* **2018**, *6*, 49546–49562. [[CrossRef](#)]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.