International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I7P104

Volume 74 Issue 7, 34-54, July 2026

Original Article

Optimal Battery Storage Sizing and Strategic Operation in Renewable Energy Integrated Grid using Roulette Chaotic Pelican Optimisation Algorithm

Ifeoluwa Titilayo Akinola[1], Yanxia Sun[2], Isaiah Gbadegesin Adebayo[3], Zenghui Wang[4]

[1,2,3]*Department of Electrical and Electronic Engineering Science, University of Johannesburg, Johannesburg 2030, South Africa.*
[3]*Department of Electronic and Electrical Engineering, Ladoke Akintola University of Technology, PMB 4000, Ogbomoso, Nigeria.*
[4]*Department of Electrical and Smart Systems Engineering, University of South Africa, Florida 1709, South Africa.*

[1]*Corresponding Author : ysun@uj.ac.za.*

Received: 21 February 2025 Revised: 08 January 2026 Accepted: 03 June 2026 Published: 28 July 2026

Abstract - *The incorporation of Renewable Energy Sources (RESs) into the power grids poses a considerable challenge due to the intermittent nature of their production. Battery Energy Storage Systems (BESS) eliminate such problems by absorbing excess generation during peaks and providing stored charge at troughs. This leads to the fact that identifying a suitable BESS size is important to ensure a trade-off between cost-efficiency and grid stability, as inappropriately sized systems would lead to inefficiency, high capital and operating costs, and reduced reliability. This study uses the Roulette Chaotic Pelican Optimisation Algorithm (RCPOA) to optimise BESS sizing in RES-integrated grids. RCPOA outperforms at least five out of seven algorithms across 23 classical benchmarks. It excels in the CEC 2019 benchmarks, ranking first in the Friedman rank test, demonstrating its robustness compared to other state-of-the-art algorithms. Its success extends to the 'tension spring design' constraint problem, underscoring its versatility in different optimisation contexts. Simulations validate RCPOA's capability to determine optimal BESS sizes, enhancing grid performance and cost efficiency. Furthermore, sensitivity analyses show that changes in State of Charge (SOC) minimum limits, battery efficiency, and range of capacities all affect the optimal battery capacity greatly. By comparison, minimum capital costs are relatively stable and are only highly sensitive to the costs of operations and maintenance, a shortfall in demand, and the cost of grid energy. These results confirm the usefulness of RCPOA to optimise BESS setup in renewable energy solutions, indicating potential to promote economic and operational goals in current power systems.*

Keywords - *Battery sizing, Energy Storage, Grid Stability, Optimisation algorithm, Algorithm modification.*

1. Introduction

The global energy industry is now in a significant change process, wherein it is motivated by the necessity to minimise emission of greenhouse gases, the need to increase energy security, as well as ensuring long-term sustainability. This move towards a cleaner energy future, as part of Sustainable Development Goal 7, requires increased incorporation of Renewable Energy Sources (RES), such as solar and wind energy, into contemporary energy infrastructures. But RES present a serious problem to grid stability and reliability [1, 2]. These challenges are mainly due to the intermittence and unpredictability of RES, which causes power supply variability. This has led to the creation of energy storage systems and especially Battery Energy Storage Systems (BESS) that form part of maintaining a balanced energy supply as excess energy is stored during times of excess generation and released during times of demand that is higher than generation [3, 4]. Battery storage systems create a flexible approach to solving the problem by being able to decouple energy supply and demand over time. They enable the levelling of the fluctuations of supply, contribute to peak shaving, improve the quality of power, and enhance smooth integration of RES [1, 5]. However, the efficiency of the battery storage towards such goals is heavily dependent on optimal sizing. Poor battery sizing causes inefficiencies, high costs, and low reliability in RES-integrated grids by either not being capable of stabilising the grid or incurring unnecessary costs through oversizing [2, 6, 7]. As such, coming up with the optimum capacity of a BESS is a very important activity, but it involves complex optimisation processes. Although meta-heuristic algorithms have become more widely used in optimising BESS sizing in the renewable-energy-integrated grids, current practices tend to fail to reach an optimal trade-off between the accuracy of the solution, the speed of convergence, and its resilience to changing grid conditions. [2]. Moreso, traditional algorithms like GA and PSO may not adequately address the dynamic and constraint-heavy nature of real-world energy systems. To bridge this gap, this study introduces an enhanced Pelican Optimisation Algorithm (POA), which enhances the original POA with tailored modifications to improve performance in BESS optimisation tasks. Pelican Optimisation Algorithm (POA) is a nature-inspired Meta-Heuristic Algorithm (MHA) that draws inspiration from pelican behaviour when hunting. It finds its application in the energy sector because of its ability to achieve the proportionate balance between exploitation and exploration necessary for data-based model optimisation, which has demonstrated its superiority over its rivals (other optimisation algorithms) [8, 9]. Since no meta-heuristic algorithm is a perfect fit for all problems according to the "No Free Lunch" (NFL) theorem, researchers are motivated to create new MHA or modify existing ones. Modifying MHA has also helped improve their performances in terms of robustness across various problem domains, computing efficiency, adaptation to restrictions, and quality of solutions [10, 11].

The novelty of this work lies in applying the enhanced POA named “Roulette Chaotic Pelican Optimisation Algorithm (RCPOA)” to achieve precise, cost-effective, and reliable battery sizing by accounting for technical constraints (such as SOC and capacity limits) and economic variables (like grid demand and energy pricing), thereby improving the integration of renewable energy sources into the power grid. By using the strengths of the RCPOA, the research aims at achieving high convergence rates, better solution quality, and increased robustness in comparison to traditional methods of optimisation. The given approach combines both technical and economic variables, such as profiles of grid loads, patterns of renewable power generation, battery State Of Charge (SOC), and electricity prices, to identify the necessary size of Battery Energy Storage System (BESS) that will help minimise all costs of the whole system and preserve grid stability and reliability. The results of the simulations and recoveries, alongside comparative studies with existing forms of optimisation, prove the effectiveness of the suggested framework in obtaining optimal BESS sizing to ensure improved optimal integration of renewable energy.

This research is part of the larger goal to increase resiliency, sustainability, and efficiency of the power grid by filling the urgent need for the development of more advanced optimisation methodologies in the energy sector. It has advanced knowledge by making significant contributions that are:

- The development of a novel Roulette Chaotic Pelican Optimisation Algorithm (RCPOA) for optimisation problems.
- Testing RCPOA on classical benchmark functions, CEC 2019 benchmark functions, and constrained engineering problems, and compared its performance with several recent algorithms.
- Formulating an optimal battery sizing model that adapts to varying renewable energy generation conditions, minimises operation costs, and ensures grid stability using historical data.
- Validating RCPOA’s performance under different grid scenarios, demonstrating its ability to optimise battery usage during peak, off-peak, and standard operating conditions, ensuring cost savings and grid stability.

The results of this research are of great relevance to the energy stakeholders, such as grid operators, policymakers, and researchers alike, thus making it easier to move to a cleaner and more reliable energy future. The study addresses one of the critical problems of renewable energy integration and grid optimisation, specifically focusing on the optimisation of battery storage systems in terms of optimal sizing through an improved optimisation algorithm. The following sections provide a background report of optimal battery sizing in the literature (Section 2), further explain on the methodology involved in the incorporation of the RCPOA (Section 3), report on RCPOA performance in terms of various benchmark functions and the resultant optimisation process (Section 4) and discuss the implications of the findings to future grid operations and energy policy development (Section 5).

2. Theoretical Background

As the energy landscape evolves, the strategic sizing of battery storage plays a pivotal role in shaping resilient and efficient power systems. The size of battery storage is crucial to achieving the best outcome of energy storage systems in grid operation, cost-efficiency, providing greater reliability, increasing renewable integration, and increasing grid efficiency. When sized properly, efficient planning, fewer uncertainties, and long-term value, through the efficient usage of batteries, are realised [2, 7].

The process of optimising the size of battery storage is a complex issue, which involves a balance of many factors, some of which are energy demand trends, the generation trends of renewable sources, the constraints of the operation, and economic aspects such as the cost of installation and the cost of maintenance. Adequate sizing makes it possible to achieve a sufficient amount of energy in Battery Energy Storage System (BESS) at a cost that is both economically viable and technically feasible. The literature points out that researchers are more concerned about the optimisation of storage size, in different aspects, namely financial, technical, or both [2, 12]. Financial considerations may involve a reduction in capital expenditure, a reduction in operating costs, or maximisation of returns on investment by exploiting market opportunities. Technical considerations usually focus on stability of the grid, system reliability, and energy efficiency [13]. These considerations determine the objective function(s) used in the optimisation process, e.g., the cost reduction [14-16] or energy use maximisation, and affect the constraints to the system. There is a great variety of constraints, including the largest possible storage capacity, the charge/discharge rates, battery life, and the environmental impact [1, 15, 17]. Finally, to choose the objective functions and constraints in battery size is to select the optimum capacity of BESS in accordance with the particular objectives of the energy system that it serves, whether these are fiscal sustainability, technical, or the optimum choice of both.

Besides the formulations, the Energy Simulation Software (ESS) and optimisation algorithms also differ, with researchers offering simpler and more efficient solutions to this optimisation problem. ESS includes HOMER [18], Transient Energy System Simulation Program (TRNSYS) [7, 19], and MATLAB [1, 19], among others. Coding programs like MATLAB facilitate the development of many mainstream innovations in optimisation research, offering flexibility and customisation that other methods, such as HOMER, lack due to their rigidity. [7, 11]. Optimisation techniques, such as Linear Programming (LP), Non-Linear Programming (NLP), Mixed Integer Linear programming (MILP) [20], Genetic Algorithms (GA), Particle Swarm Optimisation (PSO) [21], Differential Evolution (DE), Grey Wolf Optimisation (GWO) [17], and hybrid meta-heuristic algorithms [22], have been employed to tackle this problem with varying degrees of success. Researchers discussed further comparisons between optimisation processes [7, 11]. The choice of an optimisation algorithm depends on the specific properties of the problem to be solved, such as whether the objective function is linear or non-linear, whether it has discrete decision variables, and how complex the imposed constraints are. In the present study, we used a new Roulette Chaotic Pelican Optimisation Algorithm (RCPOA) in MATLAB due to its flexibility, wide-range search ability, and its ability to balance exploration and exploitation.

2.1. The Pelican Optimisation Algorithm (POA)

Recently, nature-inspired algorithms like the POA have gained attention because of their ability to search and exploit the search space at the same time, and thus, provide promising opportunities for better optimisation outcomes. The POA is an optimisation method based on the foraging behaviour and tactical choices of pelicans, which is a population-based approach. Every member of the population is called a solution candidate [8, 23]. The sampling of the individuals is random but within the constraints of the decision variables, as expressed by Equation (1).

$$x_{i,j} = I_j + rand\left(u_j - l_j\right), \mathrm{i} = 1,2,3,4,\dots,m,N;\ j = 1,2,3,4,\dots,m \quad (1)$$

Where lj is the jth lower bound, uj is the jth upper limit of problem variables, xi,j is the value of the jth variable indicated by the ith candidate solution, m is the number of problem variables, N is the number of members, and rand is a random number in the range [0, 1].

Using the provided population matrix in Equation (2), the algorithm's population members are identified. Each row in this matrix indicates a potential solution, and the columns stand for suggested solutions for the problem variables.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_j \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} X_{1,1} & \dots & X_{1,j} & \cdots & X_{1,m} \\ \vdots & \ddots & \vdots & . & \vdots \\ X_{i,1} & \cdots & X_{i,j} & . & X_{i,m} \\ \vdots & . & \vdots & \ddots & \vdots \\ X_N & \cdots & X_{N,j} & \cdots & X_{N,m} \end{bmatrix}_{N\times m} \quad (2)$$

X is the algorithm's population matrix, and X_j is the ith member.

The problem's objective function can also be assessed based on each potential solution. The values for the objective function, as given in Equation (3), are evaluated using the objective function vector.

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_i \\ \vdots \\ V_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} V(X_1) \\ \vdots \\ V(X_i) \\ \vdots \\ V(X_N) \end{bmatrix}_{N\times 1} \quad (3)$$

X_i is the value of the objective function, and V is the objective function vector of the *i*th candidate solution.

The exploration power of POA lies in the random generation of the prey location in the search space scanning. It enables POA to search the domain of potential solutions precisely. The new status of the *i*th member in the *j*th dimension based on the exploration phase $x_{i,j}^{p_1}$ is given in Equation (4);

$$x_{i,j}^{p_1} = \begin{cases} x_{i,j} + rand\ \left(p_j - I.x_{i,j}\right), & V_l < V_i \\ x_{i,j} + rand\ \left(x_{i,j} - p_j\right), & else, \end{cases} \quad (4)$$

Where 'I' is a random number (I = 1 or 2), p_j is the prey's location in the jth dimension, and V_l is the value of its objective function.

In an effective updating, a new position of the pelican can be assumed when the value of the objective function has been improved. It is mathematically explained in Equation (5).

$$X_i = \begin{cases} X_i^{p_1}, \ V_i^{p_1}, < V_i; \\ X_i, \qquad else, \end{cases} \quad (5)$$

Where $X_i^{p_1}$ is the *i*th pelican's new status, and $V_i^{p_1}$ is the value of its exploration phase objective function.

POA looks at the locations around the member position to converge on a better solution; this is where its exploitation power lies. This is done by mimicking the hunting behaviour of pelicans. It is expressed mathematically in Equation (6).

$$x_{i,j}^{p_2} = x_{i,j} + R.\left(1 - \frac{t}{T}\right).(2.rand - 1).x_{i,j} \tag{6}$$

Where $x_{i,j}^{p_2}$ is the *i*th pelican's new position in the *j*th dimension based on the exploitation phase, *T* is the maximum number of iterations, and *R* is an iteration counter.

The "$R.\left(1 - \frac{t}{T}\right)$" constant is the neighbourhood radius of $x_{i,\ j}$ (equal to 0.2). Meanwhile, 't' is the radius of the neighbourhood of the population members for local searches near each member.

At the exploitation phase, effective updating has also been employed to reject or accept the new position of the pelican. This enables POA to converge to solutions that are equal or closer to the global optimal based on Equation (7).

$$X_i = \begin{cases} X_i^{p_2}, & V_i^{p_2} < V_i \\ X_{i,} & else, \end{cases} \tag{7}$$

Where $X_i^{p_2}$ the *i*th pelican's new status and $X_i^{p_2}$ is its exploitation phase objective function value. It suggests a value for the variables in the optimisation problem based on where they are in the search space.

Despite the effectiveness of POA, its solution is stochastic. This suggests trapping in local optima and premature convergence in high-dimensional problems.

Owing to this fact, this research contributes to knowledge by enhancing POA performance by using the sinusoidal chaotic map to improve the exploration phase and introducing roulette selection to generate the prey location rather than randomly generating it.

2.2. Algorithm Modification

Several approaches have been employed to enhance meta-heuristic algorithms' performance, such as the introduction of stochastic and mathematical operations or local search, applying the chaotic maps [24, 25], tuning parameters, hybridising algorithms [26, 27], among others [28]. These processes give birth to algorithms such as Improved Chimp Optimization Algorithm [29], Modified Whale Optimisation Algorithm [30, 31], Improved sandcat swarm optimisation algorithm [32], Modified PSO [33, 34], Chaotic marine predators algorithm [35], Improved group teaching optimisation algorithm [36], A binary chaotic horse herd optimisation algorithm [37] and so on. Chaotic maps are a popular and effective way of modifying nature-inspired algorithms [38].

It is interesting to note that some variants of POA were found in a few pieces of literature focusing on improving convergence speed, accuracy, and the ability to escape local optima: The Hybrid Multi-Strategy POA (HMS-POA) integrates dynamic reverse learning, chaotic mapping, and the Harris Hawk algorithm to optimise benchmark functions and classification tasks [39].

An Improved POA with Chaotic Interference and Elementary Functions incorporates ten chaotic factors and six elementary functions, enhancing optimisation efficiency in engineering design problems [40]. The emphasis was on implementing basic mathematical functions to improve the exploitation phase.

The Mixed Strategy-Based Improved POA (MSPOA) applies Elite Opposition-Based Learning, Lévy flight, and adaptive mutation to outperform conventional meta-heuristic algorithms [41]. The Guided Pelican Algorithm (GPA) improves global and local searches by dynamically adjusting the search space and utilising multiple candidates, resulting in a performance that was 1% worse than the POA [42].

Furthermore, the Multi-Strategy Improved POA (MPOA) employs a chaotic map, Lévy flight, and Gaussian mutation mechanism and effectively applies it for mobile robot path planning and image enhancement [43, 44]. The Multi-Objective POA (MOPOA) extends POA for dynamic reconfiguration of multi-type rural rooftop PV arrays [45] and for engineering designs [46].

The Hyperchaotic POA (HCPOA) leverages hyperchaotic sequences to improve solution quality and escape local optima [47]. The Hybrid Pelican-Particle Swarm Optimisation (HPPSO) combines the POA and Particle Swarm Optimisation (PSO) to increase exploring capacity [48]. Lastly, there is the Improved Pelican Optimisation Algorithm (IPOA) that was used to address the issue of load dispatch problems [49].

Despite these research efforts, the POA still has a significant amount of potential for being optimised further by focusing on its limitations through techniques like chaotic mapping and adaptive mechanisms, as well as to test its applicability in a wide range of fields to obtain stronger, more established, and efficient optimisation solutions. In line with this, this paper presents the modification of the POA through a Roulette Wheel Selector and a sinusoidal chaotic map to optimise the sizing of the Battery Energy Storage System (BESS) in an electric grid system with a renewable energy integrated with it.

2.2.1. Chaotic Maps

The use of chaotic sequences also increases the performance of the algorithms by providing pseudo-random numbers controlled by the initial seed states and the inherent deterministic trends, in contrast with the traditional stochastic numbers [50]. Chaotic maps can either be one-dimensional, e.g., sinuoidal, circular, cubic, and logistic maps, or two-dimensional maps, e.g., the cat map, the baker map, and the Zaslavskii map.

These maps are nonlinear deterministic functions which are sensitive to the initial conditions and have been widely used to enhance the searchability of optimisation algorithms. It is expected that using a chaotic function map to alter the POA will produce a number of performance gains, including the improvement of exploration, quicker convergence, and sensitivity of parameters. Although there may be a risk of premature convergence, this can be mitigated using a suitable chaotic map and setting the parameters carefully [51].

2.2.2. Roulette Wheel Selection

Furthermore, the fitness value of solutions obtained by POA is the usual way to implement selection, which can exclude promising solutions. The introduction of a selector that will enhance this algorithm is essential.

RWS is stochastic and more straightforward to implement without discarding any candidate. The RWS method aims to mimic the process of natural selection, where fitter individuals are more likely to survive and pass their genes to the next generation.

It is similar to the literal wheel used in the game of roulette, as shown in Figure 1, which predicts where the ball will finally end. It creates a process like roulette where each candidate's fitness scores are equivalent to areas (the more significant the fitness value, the larger the area), as shown in Figure 2.

The final selection is done by rotating the roulette and picking the candidate where it points when it stops, which is more effective because it is free from biases and preserves population diversity [52-54]. Roulette wheel selection is a widely used selection method for genetic and evolutionary algorithms [8], and it can also be used in metaheuristic algorithms [55] like the Pelican Optimisation Algorithm (POA). Modifying the selection operator of POA with roulette wheel selection can affect the algorithm's performance. Consequently, thorough consideration of the specific modifications is required for the best outcomes.

After careful consideration, RWS was utilised to generate prey locations to enhance POA's performance. The selection operator is an essential component of POA, and modifying it can have a significant effect on the algorithm [34, 38, 53].

Fig. 1 Roulette wheel for the game

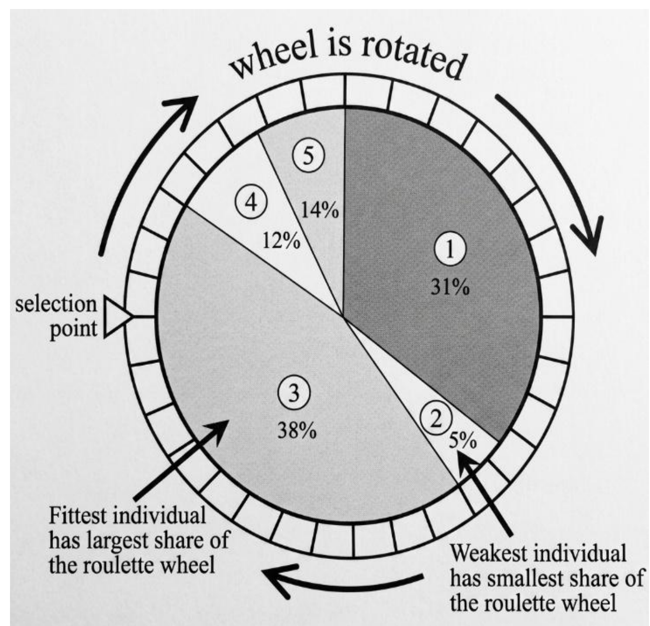


Fig. 2 Roulette wheel selection strategy [56]

3. Methodology

3.1. Roulette Chaotic Pelican Optimisation Algorithm

The Roulette Chaotic Pelican Optimisation Algorithm (RCPOA) was developed by introducing RWS to select the prey location and a sinusoidal chaotic map to the exploration phase of POA.

Since the prey location in POA is generated randomly, this research uses a roulette wheel to generate the prey location, as shown in Equation (8).

$$p_j = rand \leq \frac{f(F_i)}{\sum_{i=1}^{N} f(F_i)} \quad (8)$$

Application of RWS to generate the prey location involves the following steps:

- Calculate each candidate solution's fitness value.
- Compute the selection probabilities for each candidate solution based on their fitness values. The selection probability was proportional to the fitness value, and it determined the likelihood of a candidate solution being selected for the next generation.
- Take a random number (between 0 and 1), and use it to select the candidate solutions for the next generation. The random number was used to spin the roulette wheel, where the selection probabilities corresponded to the width of the wheel's sectors. The candidate solutions were selected based on the sector in which the random number fell.
- Apply crossover and mutation operators to produce new candidate solutions from the selected ones. Crossover combines the characteristics of two or more candidate solutions to create a different one, while mutation modifies a candidate solution randomly to introduce diversity.
- Evaluate fitness: Evaluate the fitness of the new candidate solutions using the fitness function.
- Update the population: By selecting the best candidate solutions from the current and new populations and replacing the worst ones.
- Repeat steps 'd' to 'f' for a certain number of iterations or until a termination criterion is met.

The sinusoidal map gives an excellent outcome for applications for initial population and mutation operation compared with others in the literature; hence, it is utilised for this research.

It is defined by Equation (9):

$$x_{k+1} = a x_k^2 \sin(\pi x_k) \quad (9)$$

Where a in this case is 2.3; $x_k = \frac{mod(x_{i,j}, rand)}{rand}$;

$$Chaos = x_{k+1} \times x_{i,j} \quad (10)$$

A chaotic map is applied to the exploration phase by replacing rand in Equation (4) with chaos in Equation (10) to give Equation (11).

$$x_{i,j}^{P_1} = \begin{cases} x_{i,j} + Chaos.\,(p_j - I.\,(x_{i,j})), & F_p < F_i \\ x_{i,j} + Chaos.\,(x_{i,j} - p_j), & else, \end{cases} \quad (11)$$

Figure 3 depicts the steps to arrive at the approximate optimal solution to the problem using RCPOA.

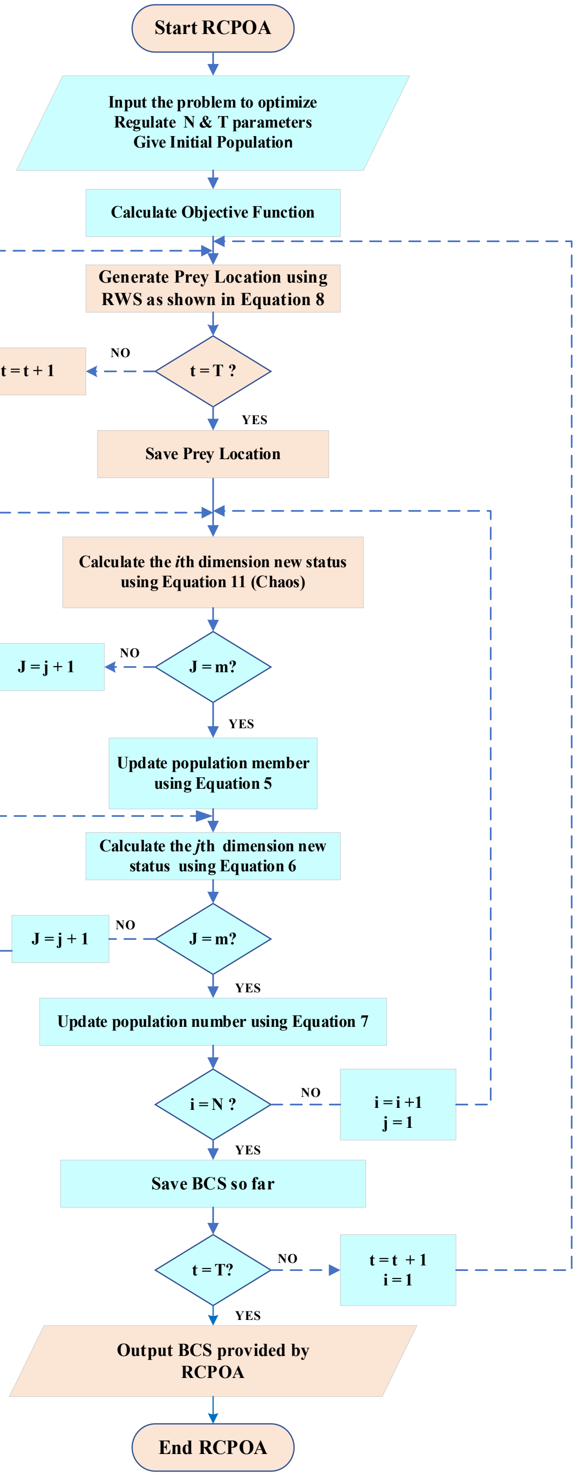


Fig. 3 Flowchart for RCPOA

The steps can be further explained as follows:

- Initialisation stage: The parameters N and T are set accordingly. The initial population is produced randomly, where each solution is represented as a set of variables or parameters.

- Evaluate fitness: This was done for each candidate solution using the fitness function.
- Apply roulette wheel selection: Select candidate solutions for the next generation by applying RWS and save prey location.
- Apply chaotic function map and calculate new status: Apply the sinusoidal map function to the candidate solutions to modify their positions in the search space. The chaotic map was used to add randomness to the search process and enhance exploration.
- Evaluate fitness: The fitness of the modified candidate solutions was evaluated using the fitness function.
- Update the population: This was done by selecting the BCS and replacing the worst ones.
- Repeat steps 4 to 6 for a certain number of iterations or until a termination criterion is met.

3.1.1. Solving unconstraint optimisation problem with RCPOA

The RCPOA was tested using twenty-three unconstrained benchmark functions (23BF) V1 –V23. The functions were divided into unimodal (V1-V8), "high-dimensional multimodal" (V8–V13), and "fixed-dimensional multimodal functions" (V14-V23), respectively. The definition of each function, the range, and the dimension used are shown in Table 1. It was also tested on CEC 2019 benchmark functions (C1-C10) with the function description shown in Table 2.

3.1.2. Solving constraint optimisation problem with RCPOA

Optimisation principles are fundamentally important in various ways in modern engineering design and system operations. RCPOA was used to solve the tension/compression spring design problem, a real-world constraint optimisation problem. Its performance was also compared with other algorithms whose results were found in the literature, such as PSO, GA, GSA, and so on. Tension/compression springs are one of the most commonly used springs in engineering to lock specific mechanical parts into a fixed position. They have found applications in lever, interior, and exterior parts of automobiles, farm machinery, and robotics, among others. Manufacturers ensure that the springs are designed in a way that they can hold components together and perform at their best. The schematic drawing of the tension/compression spring design is shown in Figure 4. The primary objective of the optimisation problem is to find the minimum weight of the Spring with the optimum value of parameters such as the diameter of the wire (d), the diameter of the coil (D), and the number of coils active in the Spring (P). This is subject to constraints like surge frequency, shear, coil diameter limits, and minimum deflection. The Optimisation problem is represented mathematically as:

Minimise the function $f(x)$ considering the continuous variables $x_1 = d, x_2 = D, x_3 = P$

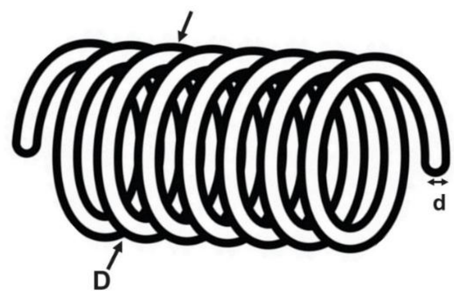
$$f(x) = (x_3 + 2)x_1^2 x_2$$

For limiting Constraints:

$$\begin{cases} g_1(x) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \leq 0, \\ g_2(x) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108 x_1^2} - 1 \leq 0, \\ g_3(x) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \leq 0 \\ g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0, \end{cases}$$

$$with\ 0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15.$$

a)



b)

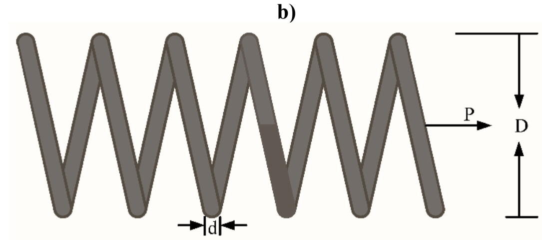


Fig. 4 Schematic drawing of compression spring design 3D representation b) 2D representation

Table 1. Defining objective functions V1-V23 [8]

Objective Function	Range	Dimensions	V_{min}	Global Optimum
$V_1(x) = \sum_{i=1}^{m} x_i^2$	[-100, 100]	30	0	0
$V_2(x) = \sum_{i=1}^{m} \lvert x_i \rvert + \prod_{i=1}^{m} \lvert x_i \rvert$	[-10, 10]	30	0	0
$V_3(x) = \sum_{i=1}^{m} \left(\sum_{j=1}^{i} x_i \right)^2$	[-100, 100]	30	0	0
$V_4(x) = max\{\lvert x_i \rvert, 1 \leq i \leq m\}$	[-100, 100]	30	0	0
$V_5(x) = \sum_{i-1}^{m-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]	30	0	0

$V_6(x) = \sum_{i=1}^{m}(\lvert x_i + 0.5\rvert)^2$	[-100, 100]	30	0	0
$V_7(x) = \sum_{i=1}^{m} ix_i^4 + random(0,1)$	[-1.28, 1.28]	30	0	0
$V_8(x) = \sum_{i=1}^{m} -x_i \, sin\left(\sqrt{\lvert x_i\rvert}\right)$	[-500, 500]	30	-12,569	-4189.8
$V_9(x) = \sum_{i=1}^{m}[x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12, 5.12]	30	0	0
$V_{10}(x) = -20exp\left(-0.2\sqrt{\frac{1}{m}\sum_{i=1}^{m} x_i^2}\right) - exp\left(\frac{1}{m}\sum_{i=1}^{m}\cos(2\pi x_i)\right) + 20 + e$	[-32, 32]	30	0	0
$V_{11}(x) = \frac{1}{4000}\sum_{i=1}^{m} x_i^2 - \prod_{i=1}^{m}\cos\left(\frac{x_i}{\sqrt{1}}\right) + 1$	[-600, 600]	30	0	0
$V_{12}(x) = \frac{\pi}{m}\{10\sin \pi y_1 + \sum_{i=1}^{m}(y_i - 1)^2[1 + 10sin^2(\pi y_{i+1})] + (y_n - 1)^2\} - \sum_{i=1}^{m} u(x, 10, 100, 4)$ Where *u (x, a, i, n)* = $\begin{cases} k(x_i - a)^n & x_i > -a \\ 0, & -a \le x_i \le a; \\ k(-x_i - a)^n & x_i < -a \end{cases}$	[-50, 50]	30	0	0
$V_{13}(x) = 0.1\{sin^2(3\pi x_i) + \sum_{i+1}^{m}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_m)]\} + \sum_{i+1}^{m} u(x, 5, 100, 4)$	[-50, 50]	30	0	0
$V_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	[-65.53, 65.53]	2	0.998	1
$V_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	[-5, 5]	4	0.00030	0.0003
$V_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5, 5]	2	-1.0316	-1.0316
$V_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	[-5, 10] × [0, 5]	2	0.398	0.398
$V_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)].[30 + (2x_1 - 3x_2)^2.(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	[-5, 5]	2	3	3
$V_{19}(x) = -\sum_{i=1}^{4} c_i \, exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - P_{ij})^2\right)$	[0, 1]	3	-3.86	-3.86
$V_{20}(x) = -\sum_{i=1}^{4} c_i \, exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - P_{ij})^2\right)$	[0, 1]	6	-3.22	-3.22
$V_{21}(x) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0, 10]	4	-10.1532	-10.1532
$V_{22}(x) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0, 10]	4	-10.4029	-10.4028
$V_{23}(x) = -\sum_{i=1}^{10}[(X - a_i)(X - a_i)^T + 6c_i]^{-1}$	[0,10]	4	-10.5364	-10.5363

Table 2. CEC 2019 benchmark functions defined

Functions	Range	Dimensions	C_{min}
C1- Storn's Chebyshev Function	[-8192, 8192]	9	1
C2- Hilbert function	[-16384,16384]	16	1
C3- Lennard_Jones Function	[-4, 4]	18	1
C4- Rastrigin's Function	[-100, 100]	10	1
C5- Griewangk's function	[-100, 100]	10	1
C6- Weierstrass' function	[-100, 100]	10	1
C7- Modified Schwefels function	[-100, 100]	10	1
C8- Expanded Schaffer's function	[-100, 100]	10	1
C9- Happy Cat function	[-100, 100]	10	1
C10- Ackley's Function	[-100, 100]	10	1

3.2. Battery Sizing Critical Parameters and Formulations

Several factors must be balanced to determine the ideal battery capacity for a Battery Energy Storage System (BESS) in an electric grid with integrated Renewable Energy Sources (RES) and to maximise grid efficiency. These factors include the variability of renewable generation, grid demand, cost, and the required degree of grid stability. On the other hand, battery operation requires BESS charging and discharging operations in response to the nature of demand and generation.

3.2.1. Critical Parameters

The critical parameters considered for this research are expressed in short forms as;

- EBESS, max: Maximum battery energy capacity (to be determined).
- PRES(t): Power the RES generates at time 't'.
- Pload(t): Power demand of the grid at time 't'.
- Pgrid(t): The Power drawn from or supplied to the grid at time 't'.
- Δt: Time step (e.g., 1 hour).
- T: Total number of time steps in the analysis period (e.g., 24 hours for a day).
- λ(t): Energy price at time 't'.
- SOC: State of charge of the battery

3.2.2. Problem formulation

The goal of the formulation is to maximise grid efficiency with the use of BESS. This will involve minimising the total cost (the cost of BESS maintenance, operations, and energy the BESS draws from the grid) while ensuring the grid remains stable. The objective function is given by Equation (12)

$$Minimise\ \sum_{t=1}^{T} C_{O\&M} + \left[\lambda(t).P_{grid}\,(t)\right] + \alpha.\sum_{t=1}^{T}\left(unstable(t)\right)^2 \tag{12}$$

$where,\ Unstable\ (t) = P_{RES}(t) + P_{BESS,discharge}(t) - P_{load}(t) - P_{BESS,charge}(t)$, α is a weighting factor that prioritizes grid stability, and $C_{O\&M}$ is the operation and maintenance cost. To get the optimal battery capacity (E_{BESS}), Equation (13) was solved subject to power balance, charging and discharging limits, and SOC dynamics and limits as shown in Equations (8)-(14).

$$Minimise\ \sum_{t=1}^{T} C_{O\&M} + \left[\lambda(t).P_{grid}\,(t)\right] + \alpha.\sum_{t=1}^{T}\left(P_{RES}(t) + P_{grid}(t) - P_{load}(t)\right)^2 \tag{13}$$

Power Balance Constraint

The power balance constraint is given by Equation (14).

$$P_{load}\,(t) = P_{RES}(t) + P_{grid}(t) + P_{BESS}(t) \tag{14}$$

Where $P_{BESS}(t) = -P_{BESS,discharge}(t)\ or + P_{BESS,charge}(t)$ for power exchange when the battery is discharging or charging, respectively.

Charging and Discharging Power Limits

The battery charging and discharging limits are represented by Equations (15) and (16), respectively.

$$0 \le P_{BESS,charge}(t) \le P_{BESS,charge,max} \tag{15}$$

$$0 \le P_{BESS,discharge}(t) \le P_{BESS,discharge,max} \tag{16}$$

SOC Constraint

The BESS State of Charge (SOC) and charging and discharging constraints were incorporated. It is given as;

$$SOC(t+1) = SOC(t) + \left(\eta_{charge}.P_{BESS,charge}(t) - \frac{P_{BESS,discharge}(t)}{\eta_{discharge}}\right).\Delta t \tag{17}$$

$$\text{For } 0 \le SOC(t) \le E_{BESS,max} \tag{18}$$

Charging and Discharging efficiency $(\eta_{charge}/\eta_{discharge})$ = 90% for this research purpose

Battery Capacity Limits

$$SOC_{min} \le SOC(t) \le SOC_{max} \tag{19}$$

SOC_{min} is 10% for this research purpose

Grid Stability Constraint

$$P_{RES}(t) + P_{BESS}(t) - P_{load}(t) \le \epsilon \tag{20}$$

Where ϵ is a small allowable imbalance to account for minor deviation.

3.3. Research Case Study

The South African historical data for generation (including renewable and non-renewable sources), load demand, and electricity tariff for 2022, as supplied by Eskom, was used for this study. Figure 5 shows the total installed capacity of the primary energy sources as reported by CSIR [57].

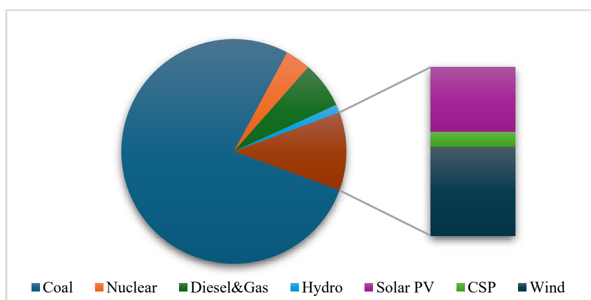


Fig. 5 Diagram showing the primary sources of South Africa's energy in 2022

The Renewable energy capacity considered for battery storage operation includes solar PV, CSP, and wind energy, making up to 11% of the mix. Although the installed total capacity of these three sources is slightly above six thousand MW, the maximum generation recorded in 2022 is about five thousand MW; this could be due to many factors. Therefore, this study used these three sources' reported hourly RES generation as the total RES power generation, other than the installed capacity. The historical data were also used to calculate the maximum battery capacity, essential for determining the optimal battery size. Eskom tariff schedules for weekdays, weekends, and holidays were divided into "Time of Use (TOU)": off-peak, standard, and peak hours. There is also an emphasis on high and low-demand seasons. Low-demand seasons are September to May, while June to August (Winter) were tagged as high-demand seasons. Table 3 shows the TOU for a day during low and high-demand season that falls on a weekday, Saturday, and Sunday. Table 4 shows the active energy charge excluding losses in (C/KWh) from Eskom. The deficit energy for Eskom's dispatchable generation and load demand is shown in Figure 6. This explicitly represents the demand RES should supply.

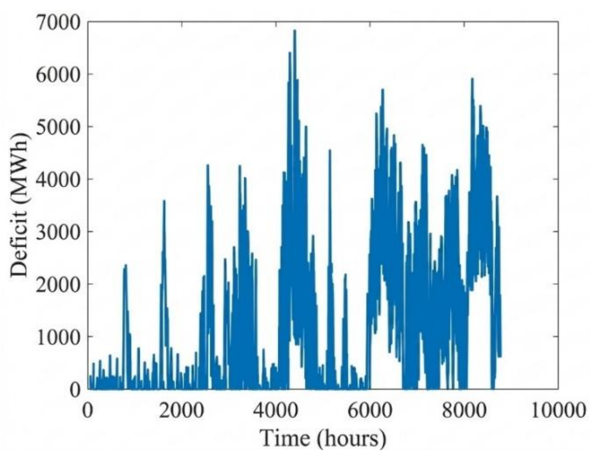


Fig. 6 Chart of load deficit from Eskom dispatchable sources in 2022

3.4. Sensitivity Analysis

After determining the optimal $E_{BESS,\ max}$, it is essential to conduct a sensitivity analysis to see how parameter changes (like RES variability, grid demand, and energy prices) affect the optimal battery capacity. This helps in ensuring robustness in the chosen capacity. Therefore, three scenarios were considered, as explained below:

- Days of low deficit: Represented by the lowest deficit value recorded in Summer, an off-peak tariff season, on February 7, 2022. (It falls on a Monday, representing a working day which records the highest peak hours according to Eskom)
- Days of average deficit: Represented by the average deficit value recorded in Spring, an off-peak tariff season, on November 14, 2022. (It also falls on a Monday)
- Days of high deficit: (Represented with the Highest deficit value recorded in Winter, a peak tariff season, July 4, 2022. (It also falls on a Monday.)

The procedure for finding the optimal battery size using the RCPOA model can be summarised as follows;

- Defining the objective function as shown in Equation (13)
- Setting constraints as shown in Equations (14)-(20)
- Collecting and preparing relevant data as discussed in Section 3.3
- Initialising RCPOA Parameters, evaluating initial Solutions, updating Pelican Positions, enforcing constraints, and ensuring that new solutions remain within defined limits, as shown in Figure 3
- Checking for convergence based on the maximum number of iterations reached
- Choosing the battery size corresponding to the best solution (lowest cost) found by RCPOA, and analysing the optimal solution's performance
- Comparing RCPOA's performance with results from other algorithms (e.g., PSO, GA, POA)
- Performing a sensitivity analysis as discussed in Section 3.4

4. Results and Discussion

The coding and simulation were performed using MATLAB ® 2023a on a laptop with Intel® Core ™ i5-6200u CPU @ 2.30GHz and 8 GB RAM.

4.1. RCPOA to Solve Unconstrained Optimisation Problem

The number of iterations used for the simulation is 1000, while the population size is 30. Two sets of unconstrained optimisation problems were considered: the classical benchmark functions (V1-V23) and the CEC 2019 benchmark functions, as explained in Section 3.1. The results obtained are presented and discussed in subsections 4.1.1 and 4.1.2.

4.1.1. Classical Benchmark Function

The Best Candidate Solution (BCS) and the average value obtained after 20 independent runs for POA and RCPOA were recorded in Table 5.

Results of other algorithms, namely: “Gravitational Search Algorithm” (GSA), PSO, Genetic Algorithm (GA), GWO, “Marine Predators Algorithm” (MPA), TLBO, and Whale Optimisation Algorithm (WOA), used to compare RCPOA, were obtained from the literature [8].

Table 3. Season and day type TOU specifications

Time	Low-demand seasons			High-demand Seasons		
	Working Days	Saturdays	Sundays	Working Days	Saturdays	Sundays
0	O	O	O	O	O	O
1	O	O	O	O	O	O
2	O	O	O	O	O	O

3	O	O	O	O	O	O
4	O	O	O	O	O	O
5	O	O	O	O	O	O
6	O	O	O	O	O	O
7	S	O	O	P	O	O
8	P	S	O	P	S	O
9	P	S	O	P	S	O
10	P	S	O	S	S	O
11	S	S	O	S	S	O
12	S	S	O	S	S	O
13	S	O	O	S	O	O
14	S	O	O	S	O	O
15	S	O	O	S	O	O
16	S	O	O	S	O	O
17	S	O	O	S	O	O
18	S	O	O	P	O	O
19	P	S	O	P	S	O
20	P	S	O	S	S	O
21	S	O	O	S	O	O
22	S	O	O	S	O	O
23	O	O	O	O	O	O

Table 4. Eskom tarrif for TOU

Cost Description	**High Demand Season**			**Low Demand Season**		
	Peak	Standard	Off-Peak	Peak	Standard	Off-Peak
Excluding VAT	543.88	164.73	89.48	177.47	122.11	77.48
Including VAT	625.46	189.44	102.90	204.09	140.43	89.10

The comparisons of these algorithms are shown in Table 5. It can be seen that from V8-V23, which are the multimodal functions, RCPOA performed relatively well in competing with other algorithms regarding exploration power. This is shown by the nearness of their BCS to the global optimum. In Table 5, it can be seen that RCPOA did better than GA in all the functions. It also performs better than PSO, TLBO, MPA, GWO, and GSA in most functions. Although it performs similarly to POA in multi-modal functions, it outperforms POA considering the average values.

To further strengthen the result obtained, another statistical analysis was performed. The "Wilcoxon Rank-Sum Test" (WRST), which is comparable in terms of its results, was employed. WRST compares two dependent samples and is non-parametric. It is used in this case to compare the performance of RCPOA with other algorithms. It determined if there was any significant difference between their performances from a statistical standpoint. The p-value and the significance level were used to determine the significant difference. Table 6 shows the p-value obtained for the comparison. The significance level (α) used was 0.05. Therefore, the null hypothesis (H_o), A=B, is rejected when p-value < α, signifying a significant difference between the two algorithms compared. Nevertheless, when the p-value is > α, the H_o is accepted, presenting no significant difference between the algorithms compared.

From the results obtained from WRST (Table 6), the p-values from these three sets of benchmark functions between RCPOA and each of the other seven algorithms provide insights into its performance. For V1-V7, several p-values (0.0156) indicate significant differences between RCPOA and others, suggesting that RCPOA performs better than many other algorithms at a statistically significant level (0.05), including POA. For V8-V13, most p-values are insignificant, with only POA (0.0312) indicating a statistically significant difference. This suggests that RCPOA's performance is similar to that of others but slightly improved compared to that of POA. For V14-V23, RCPOA shows strong performance, with p-values of 0.0117, 0.0469, 0.0078, and 0.0312, demonstrating significant improvement over GA, GSA, TLBO, and WOA. Notably, the p-value of 0.500 between POA and RCPOA in these functions implies no significant difference. Therefore, RCPOA may not substantially outperform POA for V14-V23.

Table 5. Performance of RCPOA and other algorithms with functions V1-V23

Function		GA	PSO	GSA	TLBO	GWO	WOA	POA	RCPOA
V1	BCS	5.5935	2E-10	8.2E-18	9.36E-61	7.73E-61	1.61E-65	2.53E-237	2.168E-1
	AVG	11.6208	4.1728E-4	2.03E-16	3.83E-59	1.09E-57	5.37E-62	5.52E-210	5.437E-1
V2	BCS	1.5911	0.001741	1.59E-8	1.32E-35	1.55E-35	3.42E-63	3.56E-116	4.0E-4
	AVG	4.6942	0.3114	7.06E-7	4.62E-34	2.005E-33	2.51E-55	9.99E-107	4.2E-4
V3	BCS	1014.689	1.6149	81.9124	1.21E-16	4.75E-20	1.97E-11	3.15E-233	257.215
	AVG	1361.2743	588.3012	280.6014	7.08E-14	4.72E-14	7.56E-9	8.04E-209	368.6717
V4	BCS	1.3898	1.6044	2.09E-9	6.41E-16	3.43E-16	0.0001	1.17E-116	15.0918
	AVG	2.0396	4.3693	2.63E-8	8.92E-14	1.99E-13	0.0013	8.61E-209	20.5111
V5	BCS	160.5013	3.6471	25.8381	120.7932	25.2120	26.4325	26.1969	32.8982
	AVG	308.4196	50.5412	36.0153	147.6214	27.1786	27.1754	28.2073	77.5006
V6	BCS	6	5	0	0	1.57E-5	0.01465	0	2
	AVG	15.6231	20.2691	0	0.5531	0.6518	0.0715	0	22.45
V7	BCS	0.00211	0.029593	0.01006	0.00136	0.00025	4.24E-5	9.95E-6	0.0968
	AVG	0.08652	0.3218	0.0234	0.0011	0.0077	0.00103	8.15E-5	0.1912
V8	BCS	-9717.68	-8501.44	-3969.23	-9103.77	-7227.05	-7568.9	-8783	-8199
	AVG	-8210.34	-6899.96	-2854.52	-7410.80	-5903.37	-7239.10	-7795	-6533
V9	BCS	36.8662	27.8588	4.9748	9.87396	0	0	0	19.2119
	AVG	62.1441	57.0503	16.5714	10.1379	8.10E-14	0	0	67.5248
V10	BCS	2.7572	1.1552	2.64E-9	0.156305	1.51E-14	8.88E-16	8.88E-16	2.37E-2
	AVG	3.8134	2.6301	3.54E-9	0.2691	8.62E-13	3.91E-15	3.375E-15	13.0104
V11	BCS	1.1405	7.29E-9	1.5193	0.31012	0	0	0	0.2089
	AVG	1.1973	0.0364	3.9123	0.5912	0.0013	2.03E-4	0	0.2159
V12	BCS	0.01837	0.00015	5.57E-20	0.00203	0.01929	0.00114	0.0468	4.0845
	AVG	0.0469	0,4792	0.0341	0.0219	0.0364	0.00772	0.1416	9.3329
V13	BCS	0.49809	9.99E-7	1.18E-18	0.03827	0.2978	0.02966	1.7009	15.4087
	AVG	1.2106	0.5156	0.0017	0.3306	0.5561	0.1933	2.7813	28.4213
V14	BCS	0.9980	0.9980	0.9995	0.9984	0.9980	0.9980	0.9980	0.9980
	AVG	09969	2.3909	3.9505	2.4998	4.1140	1.1061	0.9980	0.9980
V15	BCS	0.00078	0.00031	0.00081	0.00221	0.00031	0.000313	3.075E-4	3.088E-4
	AVG	0.0042	0.0528	0.0027	0.0031	0.0059	0.000463	0.0013	3.089E-4
V16	BCS	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.0316	-1.3016	-1.3016
	AVG	-1.0307	-1.0312	-1.0309	-1.0310	-1.0316	-1.0316	-1.3016	-1.3016
V17	BCS	0.3978	0.3978	0.3978	0.3978	0.3978	0.3979	0.3979	0.3979
	AVG	0.4401	0.7951	0.3980	0.3978	0.3981	0.39788	0.3979	0.3979
V18	BCS	3.0002	3	3	3	3	3	3	3
	AVG	4.3601	3.0010	3.0016	3.0010	3.0009	3.000009	3	3.0092
V19	BCS	-3.86278	-3.8627	-3.8627	-3.8625	-3.8627	-3.86278	-3.8628	-3.8628
	AVG	-3.8519	-3.8627	-3.8627	-3.8615	-3.8617	-3.86068	-3.8628	-3.8628
V20	BCS	-3.31342	-3.322	-3.322	-3.26174	-3.32199	-3.32198	-3.3220	-3.3220
	AVG	-2.8301	-3.2626	-3.0402	-3.1927	-3.2481	-3.22298	-3.3220	-3.3220
V21	BCS	-7.8278	-8.0267	-7.3506	-9.6638	-10.1532	-10.1531	-10.1532	-10.1531
	AVG	-4.2593	-5.4236	-5.2014	-9.2049	-9.6602	-8.8764	-10.1532	-10.1527
V22	BCS	-9.1106	-10.4024	-10.4026	-10.4023	-10.4021	-10.4028	-10.4029	-10.4029
	AVG	-5.1183	-7.6351	-9.0241	-10.0399	-10.4199	-9.3373	-10.1371	-10.4026
V23	BCS	-10.2227	-10.5364	-10.5364	-10.5340	-10.5363	-10.5363	-10.5364	-10.5364
	AVG	-6.5675	-6.1653	-8.9091	-9.2916	-10.1319	-9.45231	-10.5364	-10.5354

**Values presented for all algorithms were obtained from [8] except for RCPOA and POA, which were simulated*

Table 6. Wilcoxon rank test result for comparing RCPOA and other algorithms

	RCPOA/GA	RCPOA/PSO	RCPOA/GSA	RCPOA/TLBO	RCPOA/GWO	RCPOA/WOA	RCPOA/POA
V1-V7	0.2188	0.1562	0.0156	0.2188	0.0156	0.0156	0.0156
V8-V13	0.74877	0.4375	0.6875	0.1562	0.4375	0.4375	0.0312
V14-V23	0.0117	0.1250	0.0469	0.0078	0.2422	0.0312	0.500

The WRST buttress the superiority of RCPOA across the classical benchmark functions V1-V23; RCPOA demonstrates a pattern of generally better performance since POA did not have any statistical superiority to RCPOA in multi-modal functions. RCPOA is also significantly better than other algorithms, particularly GA, GSA, TLBO, GWO, and WOA.

4.1.2. CEC 2019 Benchmark functions

The CEC 2019 competition procedures were followed, and each function's average value obtained was recorded. The results obtained for POA and RCPOA are presented in Table 7, with the results of other algorithms found in the literature.

Friedman's rank test was employed, followed by the Nemenyi post hoc test to evaluate the algorithms' performance. The p-value obtained from the Friedman test for these algorithms is 1.4318e-05. This indicates that there is a significant difference among algorithms. The Friedman rank (f-rank), average rank (Ave. f-rank), Overall Rank (OR), and Rank Difference (RD) from RCPOA are also recorded in Table 7. RCPOA achieved the lowest rank (2.95), implying that it outperformed the others across the CEC 2019 benchmark functions. A further test was done using the Nemenyi post hoc test, which follows up on significant Friedman test results and calculates a Critical Difference (CD) of 3.3794. This is the threshold for determining whether rank differences between pairs of algorithms are statistically significant. The critical difference diagram is shown in Figure 7. Notably, the Nemenyi post hoc test revealed that RCPOA's rank difference with other top-performing algorithms, such as SHO, LPB, and FDO, was below the Critical Difference threshold (CD = 3.38), meaning that these slight rank differences were not statistically significant.

Table 7. Performance for CEC 2019 and Friedman rank test

Functions	SHO [25]	LPB[58]	FDO[59]	JDE100[60]	WOA[59]	PSO[58]	POA	RCPOA
C1	4.87E4	7.49E9	4.585E3	1.59E5	4.11E10	1.5E12	3.76E4	1.38E8
f-rank	3	6	1	4	7	8	2	5
C2	17.3	17.64	4.0	2.385E5	17.3495	382.33	17.34	17.34
f-rank	2	6	1	8	5	7	3.5	3.5
C3	12.7	12.7024	13.7024	1.31E5	13.7024	12.7024	12.70	12.70
f-rank	2	4.5	6.5	8	6.5	4.5	2	2
C4	124	77.90	34.0839	3.475E5	349.6754	16.8	33.89	33.62
f-rank	6	5	4	8	7	1	3	2
C5	1.69	1.188	2.13924	1.673E5	2.7342	1.1382	1.45	1.33
f-rank	5	2	6	8	7	1	4	3
C6	7.91	3.739	12.1332	3.841E4	10.7085	9.3053	7.93	7.86
f-rank	3	1	7	8	6	5	4	2
C7	7.55	145.28	120.4858	9.105E6	490.6843	160.686	160.35	90.5123
f-rank	1	4	3	8	7	6	5	2
C8	5.06	4.8876	6.1021	1.219E9	6.909	5.224	4.40	4.10
f-rank	4	3	6	8	7	5	2	1
C9	2.01	2.894	2.0	9.207E8	5.9371	2.3732	4.88	5.16
f-rank	2	4	1	8	7	3	5	6
C10	20.2	20.002	2.7182	1.541E6	21.2761	20.28	20.16	20.13
f-rank	5	2	1	8	7	6	4	3
Total f-rank	33.0	37.5	36.5	76.0	66.5	46.5	34.5	29.5
Ave. f-rank	3.30	3.75	3.65	7.60	6.65	4.65	3.45	2.95
OR	2	5	4	8	7	6	3	1
RD	0.35	0.80	0.70	4.65	3.70	1.70	0.50	

In contrast, WOA and JDE100 differ significantly from RCPOA, with a rank difference of 3.70 and 4.65, respectively. Other algorithms do not differ significantly because their rank differences are not above CD.

The Friedman test results indicate a statistically significant difference in performance among the eight algorithms, with a very low p-value (1.4318e-05), showing that the algorithms are not all performing equally well. Based on average ranks, JDE100 ranked highest (7.60) and had substantial rank differences with several algorithms, indicating significantly poorer performance.

The non-significant rank differences between RCPOA and other top-performing algorithms suggest a consistently high performance, while its rank advantage points to a slight edge over its closest competitors.

RCPOA also maintained stability across tests with a low average rank, reinforcing its reliability as the most effective algorithm among those tested. This makes RCPOA a robust choice for scenarios requiring reliable and optimal solutions in benchmark functions, as it consistently outperforms or matches the efficiency of other algorithms and notably performs better than POA.

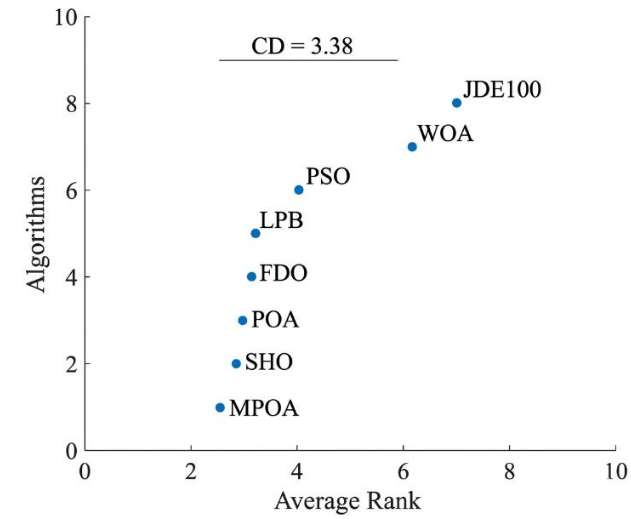


Fig. 7 CD Diagram for the algorithms' performance on the CEC 2019 benchmark functions

4.2. RCPOA for Constraint Optimisation (Tension Spring Design)

The simulation was performed 20 times independently, with 30 population numbers and 1000 iterations. The results of the optimisation problem by the proposed algorithms and other prominent algorithms found in the literature are presented in Table 8. The statistical results for further comparison are presented in Table 9. From this result, it was seen that RCPOA outperformed other algorithms with better statistical results. RCPOA gave the lowest optimum cost. It also shows more precision and accuracy, as revealed by the standard deviation, which is the lowest. Figure 8 shows the rate of the algorithm's convergence towards the best solution obtained by RCPOA for tension/compression spring design (convergence curve). With the better performance of RCPOA in constraint and unconstrained optimisation problems, it has an edge over other optimisation algorithms to solve real-world problems like the optimal battery sizing problem presented in this study.

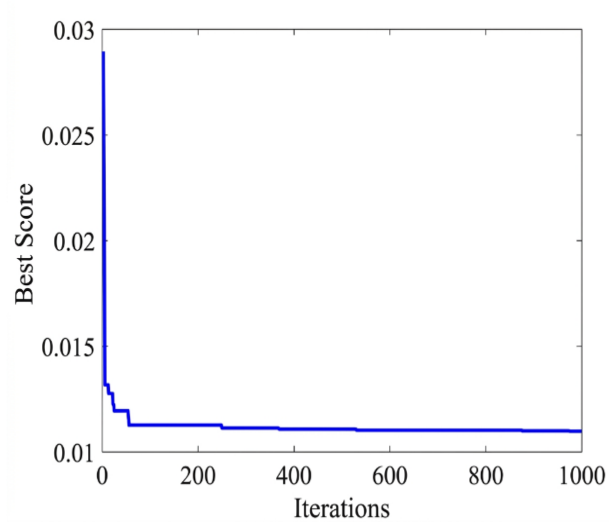


Fig. 8 RCPOA convergence curve

Table 8. Optimisation results for tension/compression spring design

Algorithm	D	D	P	Optimum cost
RCPOA	0.0500212	0.349234	10.55190	0.010968
POA [8]	0.0518637	0.360933	11.04610	0.012666
PSO [8]	0.0501100	0.310173	14.00280	0.013039
GA [8]	0.0502600	0.316414	15.24270	0.012779
GSA [8]	0.0500100	0.317375	14.23152	0.012876
GWO [8]	0.0500100	0.316019	14.22908	0.012819
WOA [8]	0.0500100	0.310476	15.00300	0.013195
PbA [61]	0.0518200	0.359887	11.10560	0.012665
SA [62]	0.0516470	0.355691	11.35190	0.012668

Table 9. Statistical report of tension/compression spring design optimisation results

Algorithm	Best	Worst	AVG	MED	STD
RCPOA	0.010968	0.011013	0.0110	0.0110	0.000011030
POA [8]	0.012666	0.012731	0.0127	0.0127	0.000003645
PSO [8]	0.013039	0.016263	0.0135	0.0130	0.002074000
GA [8]	0.012779	0.015225	0.0128	0.0130	0.000375000
GSA [8]	0.012876	0.014222	0.0135	0.0134	0.000287000
GWO [8]	0.012819	0.017852	0.0145	0.0140	0.001623000

WOA [8]	0.013195	0.017875	0.0148	0.0132	0.002274000
PbA[61]	0.012665	0.017050	0.0134	-	0.001300000
SA [62]	0.012668	0.013802	0.0128	0.0127	0.000279000

4.3. Optimal Battery Sizing using RCPOA

The maximum energy deficit from the historical data is 6835.80 MW. This was recorded on the 4th of July. The maximum rated power and the estimated Maximum Battery capacity are as follows: The P_{BESS} limit used is (6840, 7500 MWh), a range to accommodate the maximum deficit in the historical data. The $E_{BESS,\,max}$ is 800000 MWh, $E_{BESS,\,min}$ is 10% of $E_{BESS,max,}$ to cater for the cumulative annual energy deficit/ surplus. The SOC min is 10% to strengthen the battery life and battery charging and discharging efficiency of 90%, as suggested by researchers. [20, 63, 64]. The study focuses on stability for peak load and outages. Therefore, the optimal battery size will focus on the peak season. The nature of electric demand on RES and the corresponding RES generation for a peak demand season is represented by Figure 9. A damp load is introduced to help obtain the optimal BESS capacity for stability due to the nature of the load during peak and off-peak seasons.

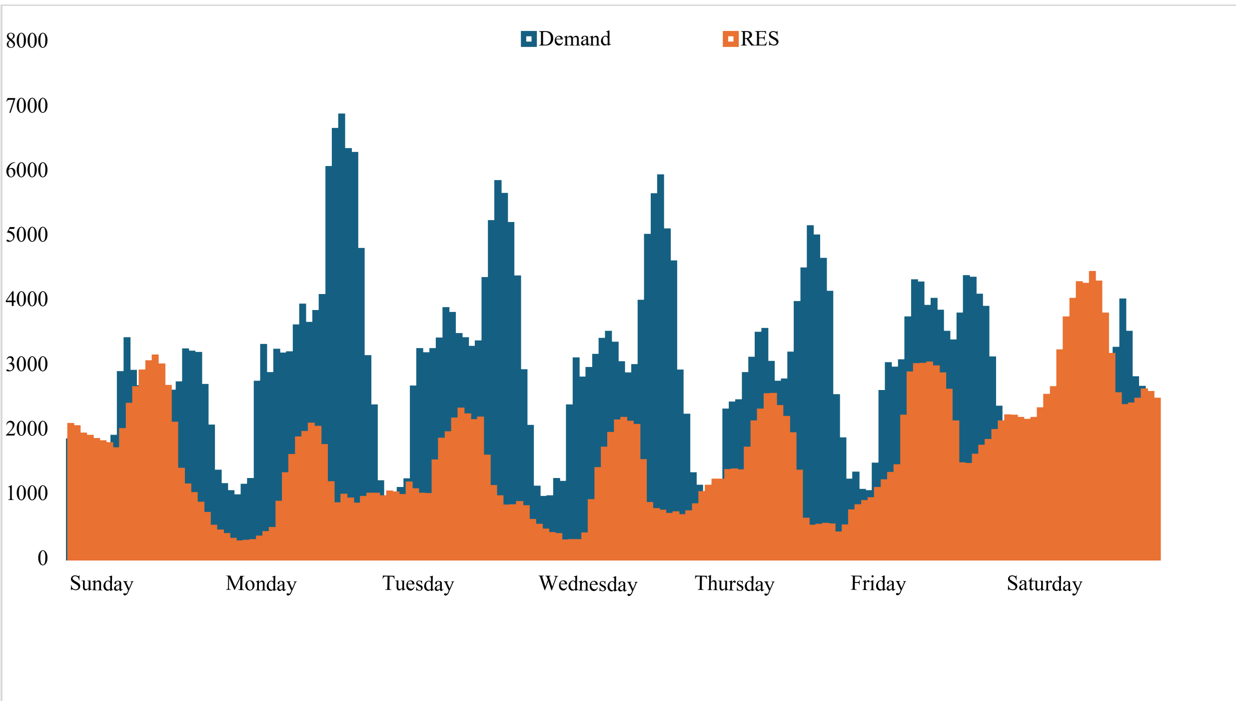


Fig. 9 Required demand and RES generation for a week in summer (July 3-9, 2022)

4.3.1. Comparing Algorithm Performance for Optimal Battery Sizing

The result obtained by RCPOA was compared with POA, PSO AND GWO. All four algorithms used 30 population numbers and 100 maximum iterations. Other PSO-specific parameters are Social and Cognitive (C1, C2) =1.5, inertia weight 0.7, and damping ratio 0.99. Due to the stochastic elements in the algorithms, 50 runs were done for each algorithm, and the 20 best results were recorded. Each algorithm's daily minimal running cost remained constant while the optimal battery size varied slightly for the same set of inputs. The mean and standard deviation obtained from the varying battery sizes are shown in Table 10. It is noteworthy that RCPOA performed outstandingly with the best solution and average. Figure 10 shows the SOC of the battery throughout the year, with an optimal size of 761000 MWh. RCPOA has the lowest cost (R417,177.91), making it the best solution in terms of cost-effectiveness. RCPOA achieves the minimum cost and suggests the smallest optimal battery size, indicating a more efficient use of battery capacity, as shown in Figure 10. The battery sizes suggested by the algorithms vary slightly, but all are within a similar range, between 765,098 and 773,952 MWh. Despite these similar results, the four algorithms explore the search space differently.

Table 10. Result for optimal battery sizing

	RCPOA	POA	GWO	PSO
Minimum cost (R)	417177.91	423301.1	563352.62	418829.88
Best Size	765098.586	773952.0796	772799.635	766025.0484

Average	763783.671	794887.5273	722170.8605	767310.858
STD	15119.7022	7330.9553	30982.8709	25466.5784
Time (s)	21.482	21.794	21.582	22.489

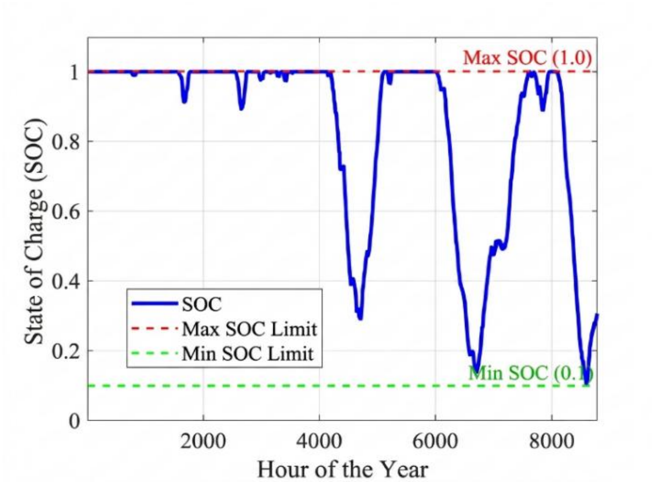


Fig. 10 State of charge of proposed battery storage for the year

Moreover, the RCPOA displayed a strong strength compared to other algorithms, and hence has more consistent results featuring a lower standard deviation, and it has an average performance that was very close to the best-known solution. This observation not only tests reliability but is also robust in handling the optimisation task.

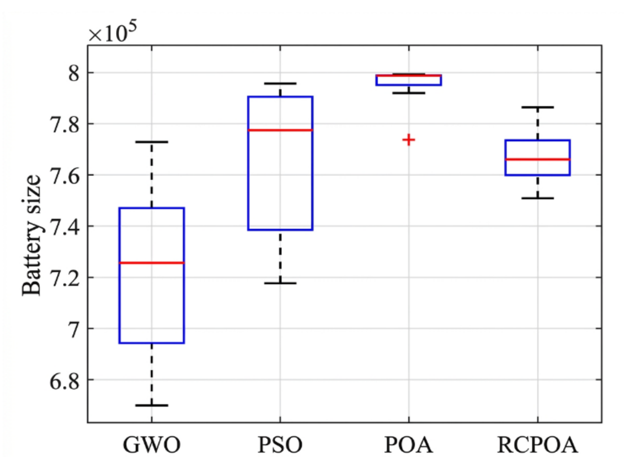


Fig. 11 Comparison of optimal battery sizes obtained

The RCPOA registered the highest mean run time of 21.68 seconds, barely beating the rest, which registered 21.794, 21.582, and 21.489, respectively. These results highlight that the RCPOA can be used to provide high-quality solutions with low computational cost. Figure 11 gives a boxplot of the optimum battery sizes of the four algorithms based on the top twenty runs, hence shedding light on the better performance of the RCPOA on the optimisation of battery size.

The RCPOA had a lower median battery size and a tighter distribution compared to the other algorithms, indicating high consistency and low variability. In contrast to the POA, which was also stable, it exhibited an inclination to oversize. The RCPOA provides a strong balance between cost-effectiveness and strength, thus presenting its ability to present viable quality and feasible solutions.

4.3.2. Sensitivity Analysis and Battery Management

Sensitivity analysis indicates the dependence of the various parameters on the strength of the BESS cost-minimisation solutions.

This was done to determine the parameters that are of utmost concern when keeping the cost minimum and the battery capacity at the optimum, and to vary the range of the system flexibility, hence enabling practical decision-making based on cost stability and flexibility.

It was analysed on variations in minimum State of Charge (SOC min), Battery Efficiency, Battery Energy Capacity (EBESS), dataset scenarios (scenario 1, 2, and 3), and running cost (operations and maintenance, O&M). Table 11 and Figure 12 represent the results obtained.

Findings show that with a rise in SOC min, an increment in the Optimum Battery Capacity (OBC) is insignificant, and the Minimum Cost (MC) is the same. This implies that cost management of O&M can play a vital role in cost optimisation in general without changing the BESS setup.

Table 11. Result of sensitivity analysis

Charging time (hours)		Battery size	Minimum cost
Change in SOC_{min}	SOC_{min} @0.1	765098.5863	417177.91
	SOC_{min} @0.15	769163.2318	417177.91
	SOC_{min} @0.2	776618.2351	417177.91
Change in Battery efficiency	0.85	776680.3045	417177.91
	0.9	765098.5863	417177.91
	0.95	762726.3652	417177.91
Change in $E_{BESS, max}$	10000	10000	12027421.81
	50000	50000	521219.12

	780000	765604.7841	417177.91
Change in Data (RES, demand, and grid cost)	Scenario 1	765484.1582	174724.00
	Scenario 2	765098.5863	417177.91
	Scenario 3	761280.1877	107597.86
Change in Operation and maintenance cost	$C_{O\&M}/2$	766020.1903	196319.02
	$C_{O\&M}$	765098.5863	417177.91
	$C_{O\&M}*2$	765297.0629	834355.83

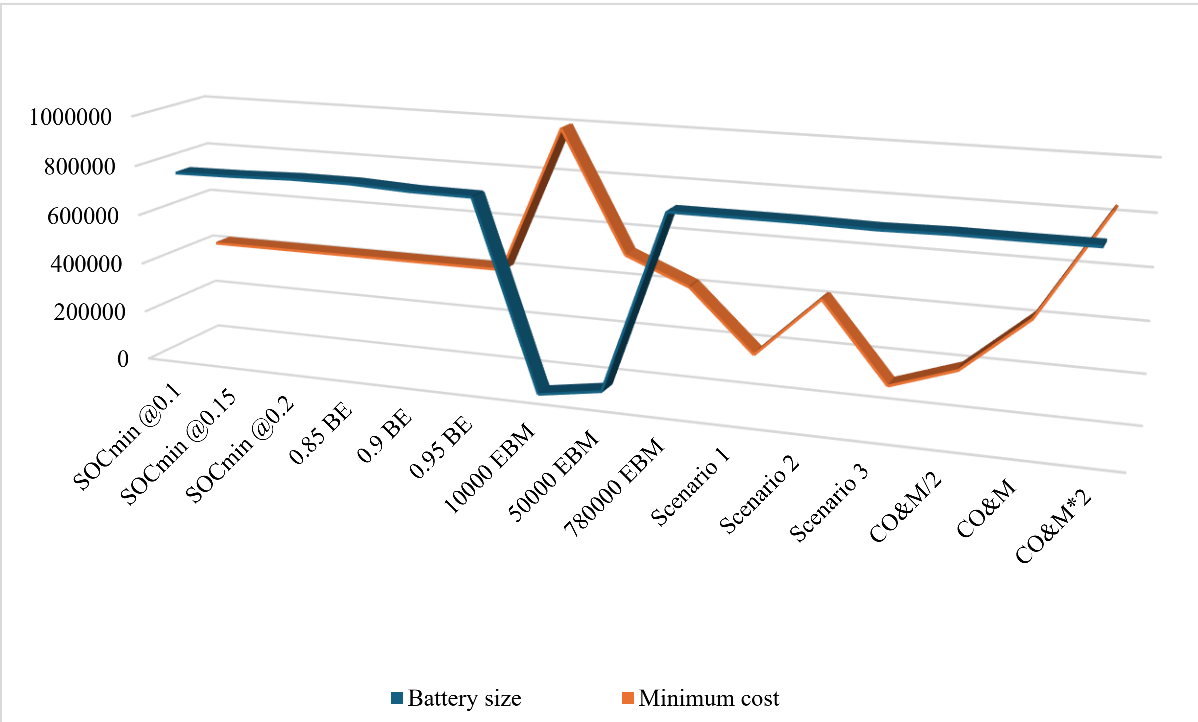


Fig. 12 Graph showing results for sensitivity analysis

The MC is unaffected because the cost structure is not sensitive to these changes, likely due to the constant energy usage pattern that the battery can support under all SOC_min scenarios. Changes in battery efficiency also alter the OBC while MC remains constant. 0.85 efficiency gave the highest OBC, while 0.95 gave the lowest. Lower efficiency means more energy is lost during charging/discharging, requiring more extensive storage to meet the exact demand and account for losses. However, MC stability implies that the cost optimisation approach effectively compensates for efficiency changes, allowing flexibility in efficiency without impacting cost. Furthermore, a change in battery capacity range significantly affects both OBC and MC. The Maximum battery capacity ($E_{BESS,\ max}$) that is very close to the OBC gave the same MC with a similar OBC, while any EBESS, max that is very low compared to OBC, returned EBESS, max as the OBC and has a high MC. The lower the $E_{BESS,\ max}$, the higher the MC. Constraining capacity below the required OBC forces the system to draw more energy from the grid, increasing costs.

Change in the dataset affects the MC significantly, with an insignificant change in OBC. Scenario 1 depicts a day of lower demand and higher RES with a lower cost of energy drawn from the grid, which gives R1747240, while scenario 2, with the highest load deficit and high energy cost, has R417177.91.

The third scenario depicts an average deficit of R107597.86. The BESS significantly reduces grid costs when renewable generation is high relative to load. Conversely, during high-demand, low-renewable periods, the system draws more from the grid, increasing costs. Proper charging and discharging schedules will help to optimise costs further. Change in $C_{O\&M}$ also affected the MC with an insignificant change in OBC. $C_{O\&M}$ gave R417177.91, $C_{O\&M}/2$ gave R196319.02 and $C_{O\&M}*2$ gave R834355.83. Findings show that with a rise in SOC min, an increment in the optimum battery capacity (OBC) is insignificant, and the minimum cost (MC) is the same. This implies that cost management of O&M can play a vital role in cost optimisation in general without changing the BESS setup.

The sensitivity analysis also indicates that Optimal Battery Capacity (OBC) depends on changes in SOC minimum limits, battery efficiency, as well as capacity range. Conversely, the least cost is not much sensitive on these parameters, and it seems to exhibit much cost sensitivity only to the change in the operational and maintenance cost and the change in the condition of grid demand. In that vein, the range of BESS capacity should also surpass the optimal battery size in order to effectively manage the minimal addition of cost to the grid, high-efficiency battery should be used in order to control the necessary capacity to be handled without inflating costs, SOCmin should be adjusted based on the operational strategies (e.g., management of the battery lifespan) as it has the least cost, BESS operation should be adjusted to periods of high renewable energy production or low cost of grid electricity to maximise the cost under different demand conditions and O&M costs should be tended to enhance the BESS's economic performance.

5. Conclusion & Future Research

In conclusion, this paper reveals the usefulness of the Roulette Chaotic Pelican Optimisation Algorithm (RCPOA) in identifying the most optimal Battery Energy Storage System (BESS) size to be incorporated in the grids that are connected to Renewable Energy Sources (RES). The RCPOA has been shown to perform exceptionally well, out of a number of well-known optimisation algorithms in constrained and unconstrained optimisation problems, possessing better and greater accuracy, strength, and versatility. The simulations and case studies showed that the RCPOA had high potential to optimise BESS to better the grid performance, reduce its operational costs, and enhance its reliability. Sensitivity analysis also showed that, although optimal battery capacity is renewable in reaction to battery efficiency, SOC constraints, and capacity spectrum, the least expenses are largely presented by operations and upkeep costs and grid requirement conditions. Such results highlight the power of RCPOA as a powerful and consistent tool to manage RES-integrated grids, in which it can help considerably reduce the energy costs and grid stability. To ensure the most out of the RCPOA, it is suggested that the algorithm be incorporated into the real-time grid control systems to dynamically modify BESS operation based on the fluctuation in the RES generation and demand. Future studies may examine the optimisation of refining the parameters of RCPOA to enhance flexibility in different grid states, especially in massive RES-integrated networks. Also, multi-objective RCPOA solutions might be added in order to optimise cost, emission, and reliability rates. Investigations into battery charging and discharging schedules, as well as the deployment of hybrid storage to reduce BESS capacity, may also be examined for practical applications.

Conflicts of Interest

The authors affirm that they have no known financial or interpersonal conflicts regarding the publication of this paper.

Funding Statement

This work was partly supported by the South African National Research Foundation under Grant nos. CPRR240327211161, CHN231229201835 and AJCR230704126719.

Authors Contribution

All authors contributed to the conceptualisation and design of the study. Simulations, data curation, and analysis were performed by I T A. The project was supervised by Y. S. and I. G. A. Y S provided funding acquisition and project resources. I T A. wrote the first draft, while Y S and Z W. thoroughly reviewed and edited it. All authors read and approved the final manuscript.

References

[1] Ruixiaoxiao Zhang et al., "Optimization of Battery Energy Storage System (BESS) Sizing in Different Electricity Market Types Considering BESS Utilization Mechanisms and Ownerships," *Journal of Cleaner Production*, vol. 470, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Yuqing Yang et al., "Battery Energy Storage System Size Determination in Renewable Energy Systems: A Review," *Renewable and Sustainable Energy Reviews*, vol. 91, pp. 109-125, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Dlzar Al kez et al., "A Critical Evaluation of Grid Stability and Codes , Energy Storage and Smart Loads in Power Systems with Wind Generation," *Energy*, vol. 205, pp. 1-22, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Amela Ajanovic, Albert Hiesl, and Reinhard Haas, "On the Role of Storage for Electricity in Smart Energy Systems," *Energy*, vol. 200, pp. 1-18, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Apar Chitransh, and Ranjit Kumar Bindal, "A Review on Energy Storage System of Smart Grid System," *International Journal of Science and Research*, vol. 10, no. 6, pp. 38-43, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[6] Olalekan Kunle Ajiboye et al., "A Review of Hybrid Renewable Energies Optimisation: Design, Methodologies, and Criteria," *International Journal of Sustainable Energy*, vol. 42, no. 1, pp. 648-684, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[7] Yijie Zhang, Tao Ma, and Hongxing Yang, "A Review on Capacity Sizing and Operation Strategy of Grid-Connected Photovoltaic Battery Systems," *Energy and Built Environment*, vol. 5, no. 4, pp. 500-516, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Pavel Trojovský, and Mohammad Dehghani, "Pelican Optimization Algorithm: A Novel Nature-Inspired Algorithm for Engineering Applications," *Sensors*, vol. 22, no. 3, pp. 1-34, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] K. Anjani Parvathi, N.C. Kotaiah, and K. Radha Rani, "Pelican Optimization Algorithm for Optimal Demand Response in Islanded Active Distribution Network Considering Controllable Loads," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 6, pp. 132-141, 2022. [CrossRef] [Google Scholar]

[10] Anuj Kumar et al., "Modified Wild Horse Optimizer for Constrained System Reliability Optimization," *Axioms*, vol. 12, no. 7, pp. 1-17, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Asmita Ajay Rathod, and S. Balaji, "Optimization of Stand-Alone Hybrid Renewable Energy System based on Techno-Socio-Enviro-Financial Perspective using Improved Red-Tailed Hawk Algorithm," *Applied Energy*, vol. 376, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Roya AhmadiAhangar et al., "Impacts of Grid-Scale Battery Systems on Power System Operation, Case of Baltic Region," *IET Smart Grid*, vol. 7, no. 2, pp. 101-119, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Saman Korjani et al., "Battery Management for Energy Communities-Economic Evaluation of an Artificial Intelligence-Led System," *Journal of Cleaner Production*, vol. 314, pp. 1-36, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Abed Kazemtarghi, and Ayan Mallik, "Techno-Economic Microgrid Design Optimization Considering Fuel Procurement Cost and Battery Energy Storage System Lifetime Analysis," *Electric Power Systems Research*, vol. 235, pp. 1-31, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Mohammed Atta Abdulgalil, Muhammad Khalid, and Fahad Alismail, "Optimal Sizing of Battery Energy Storage for a Wind Uncertainties," *Energies*, vol. 12, no. 12, pp. 1-29, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Umar T. Salman, Fahad Al-Ismail Saleh, and Khalid Muhammad, "Optimal Sizing of Battery Energy Storage for Grid-Connected and Isolated Wind-Penetrated Microgrid," *IEEE Access*, vol. 8, pp. 91129-91138, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Sharmistha Sharma, Subhadeep Bhattacharjee, and Aniruddha Bhattacharya, "Grey Wolf Optimisation for Optimal Sizing of Battery Energy Storage Device to Minimise Operation Cost of Microgrid," *IET Generation, Transmission and Distribution*, vol. 10, no. 3, pp. 625-637, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Mamood Gholizadeh, and Martin Wolter, "Cost-Beneficial Analysis of Utilizing a Combination of Renewable and Non-Renewable Energy Sources," *2020 55th International Universities Power Engineering Conference*, Turin, Italy, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[19] Polamarasetty P. Kumar, and Rajeshwer Prasad Saini, "Optimization of an Off-Grid Integrated Hybrid Renewable Energy System with Various Energy Storage Technologies using Different Dispatch Strategies," *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 47, no. 1, pp. 274-303, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[20] Mohammad Amini et al., "Optimal Sizing of Battery Energy Storage in a Microgrid Considering Capacity Degradation and Replacement Year," *Electric Power Systems Research*, vol. 195, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] M. Faisal et al., "Particle Swarm Optimised Fuzzy Controller for Charging-Discharging and Scheduling of Battery Energy Storage System in MG Applications," *Energy Reports*, vol. 6, pp. 215-228, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] MohammadReza AkbaiZadeh, Taher Niknam, and Abdollah Kavousi-Fard, "Adaptive Robust Optimization for the Energy Management of the Grid-Connected Energy Hubs based on Hybrid Meta-Heuristic Algorithm," *Energy*, vol. 235, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Qi Xiong, Jincheng She, and Jinkun Xiong, "A New Pelican Optimization Algorithm for the Parameter Identification of Memristive Chaotic System," *Symmetry*, vol. 15, no. 6, p. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] K.M. Ang et al., "Modified Particle Swarm Optimization with Chaotic Initialization Scheme for Unconstrained Optimization Problems," *Mekatronika*, vol. 3, no. 1, pp. 35-43, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Feyza Altunbey Özbay, "A Modified Seahorse Optimization Algorithm based on Chaotic Maps for Solving Global Optimization and Engineering Problems," *Engineering Science and Technology, an International Journal*, vol. 41, pp. 1-26, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Kamini Shahare, "Performance Analysis and Comparison of Various Techniques for Short-Term Load Forecasting," *Energy Reports*, vol. 9, pp. 799-808, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[27] Khaidem Bidyanath, Sanasam Dhanabanta Singh, and Shuma Adhikari, "Implementation of Genetic and Particle Swarm Optimization Algorithm for Voltage Profile Improvement and Loss Reduction using Capacitors in 132 KV Manipur Transmission System," *Energy Reports*, vol. 9, pp. 738-746, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[28] Cristina Bianca Pop et al., "Review of Bio-Inspired Optimization Applications in Renewable-Powered Smart Grids : Emerging Population-based Metaheuristics," *Energy Reports*, vol. 8, pp. 11769-11798, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Peiyao Yang et al., "Application of Improved Chimp Optimization Algorithm Integrating Multiple Strategies in Reservoirs Optimization Operation," *Expert Systems with Applications*, vol. 285, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[30] Kumeshan Reddy, and Akshay K. Saha, "A Modified Whale Optimization Algorithm for Exploitation Capability and Stability Enhancement," *Heliyon*, vol. 8, no. 10, pp. 1-11, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Vamsi Krishna Reddy Aala Kalananda, and Venkata Lakshmi Narayana Komanapalli, "A Combinatorial Social Group Whale Optimization Algorithm for Numerical and Engineering Optimization Problems," *Applied Soft Computing*, vol. 99, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[32] Heming Jia et al., "Improved Sandcat Swarm Optimization Algorithm for Solving Global Optimum Problems," *Artificial Intelligence Review*, vol. 58, no. 1, pp. 1-38, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[33] S. Vikram Singh et al., "Enhancing Microgrid Stability: Optimal BESS Sizing via Modified PSO for Frequency Control and Cost Efficiency," *2024 7th International Conference on Contemporary Computing and* Informatics, Greater Noida, India, pp. 627-632, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Sahar Saeedi et al., "Improved Many-Objective Particle Swarm Optimization Algorithm for Scientific Workflow Scheduling in Cloud Computing," *Computers and Industrial Engineering*, vol. 147, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[35] Sumit Kumar et al., "Chaotic Marine Predators Algorithm for Global Optimization of Real-World Engineering Problems," *Knowledge-based Systems*, vol. 261, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Hamed Khosravi et al., "An Improved Group Teaching Optimization Algorithm based on Local Search and Chaotic Map for Feature Selection in High-Dimensional Data," *Expert Systems with Applications*, vol. 204, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[37] Esin Ayşe Zaimoğlu et al., "A Binary Chaotic Horse Herd Optimization Algorithm for Feature Selection," *Engineering Science and Technology, an International Journal*, vol. 44, pp. 1-22, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Ruba Abu Khurma et al., "A Review of the Modification Strategies of the Nature Inspired Algorithms for Feature Selection Problem," *Mathematics*, vol. 10, no. 3, pp. 1-45, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[39] Hao Liu, Hongbin Dong, and Jing Zhou, "Feature Selection Method based on Hybrid Multi-Strategy Pelican Optimization Algorithm," *2023 6th International Conference on Data Storage and Data Engineering*, Mianyang, China, pp. 59-63, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Hao-Ming Song et al., "Improved Pelican Optimization Algorithm with Chaotic Interference Factor and Elementary Mathematical Function," *Soft Computing*, vol. 27, no. 15, pp. 10607-10646, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[41] Ziqiang Zhang, and Yixin Zhao, "Mixed Strategy-based Improved Pelican Optimization Algorithm," *2023 4th International Symposium on Computer Engineering and Intelligent Communications*, Nanjing, China, pp. 428-433, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[42] Purba Daru Kusuma, and Anggunmeka Luhur Prasasti, "Guided Pelican Algorithm," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 6, pp. 179-190, 2022. [CrossRef] [Google Scholar]

[43] Chun Qing Li, Zheng Feng Jiang, and Yong Ping Huang, "Multi-Strategy Improved Pelican Optimization Algorithm for Mobile Robot Path Planning," *Information Technology and Control*, vol. 53, no. 2, pp. 372-389, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[44] Xiaohan Zhao et al., "Enhancement Method based on Multi-Strategy Improved Pelican Optimization Algorithm and Application to Low-Illumination Forest Canopy Images," *Forests*, vol. 15, no. 10, pp. 1-30, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[45] Lingzhi Yi et al., "A Multi-Objective Pelican Optimization Algorithm for Dynamic Reconfiguration of Multi-Type Rural Rooftop PV Array," *Journal of Intelligent and Fuzzy Systems: Applications in Engineering and Technology*, vol. 47, no. 5-6, pp. 393-409, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[46] Y. Ramu Naidu, "Multi-objective Pelican Optimization Algorithm for Engineering Design Problems," *Distributed Computing and Intelligent Technology: 19th International Conference, ICDCIT 2023*, Bhubaneswar, India, pp. 362-368, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[47] Xizhai Ge et al., "A Hyperchaotic Map with Distance-Increasing Pairs of Coexisting Attractors and its Application in the Pelican Optimization Algorithm," *Chaos, Solitons and Fractals*, vol. 173, pp. 1-12, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[48] Amit Raj, Parul Punia, and Pawan Kumar, "A Novel Hybrid Pelican-Particle Swarm Optimization Algorithm (HPPSO) for Global Optimization Problem," *International Journal of System Assurance Engineering and Management*, vol. 15, no. 8, pp. 3878-3893, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[49] SeyedDavoud SeyedGarmroudi et al., "Improved Pelican Optimization Algorithm for Solving Load Dispatch Problems," *Energy*, vol. 289, pp. 1-32, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[50] Gokulakrishnan Sriram et al., "Dynamics of a Novel Chaotic Map," *Journal of Computational and Applied Mathematics*, vol. 436, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[51] Rasika B. Naik, and Udayprakash Singh, "A Review on Applications of Chaotic Maps in Pseudo-Random Number Generators and Encryption," *Annals of Data Science*, vol. 11, no. 1, pp. 25-50, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[52] Herman Herman et al., "Scheduling using Genetic Algorithm and Roulette Wheel Selection Method Considering Lecturer Time," *Journal Information of Technology and its Utilization*, vol. 2, no. 1, p. 24-28, 2019. [Google Scholar] [Publisher Link]

[53] Rosshairy Abd Rahman et al., "Evolutionary Algorithm with Roulette-Tournament Selection for Solving Aquaculture Diet Formulation," *Mathematical Problems in Engineering*, vol. 2016, no. 1, pp. 1-10, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[54] G.S. Geethamani, and M. Mayilvaganan, "Analysis of Genetic Crossover Techniques based on Roulette Wheel Selection Algorithm and Steady State Selection Algorithm," *International Journal of Engineering and Computer Science*, vol. 5, no. 1, pp. 15428-15431, 2016. [CrossRef] [Google Scholar]

[55] Asaad Shakir Hameed et al., "Appling the Roulette Wheel Selection Approach to Address the Issues of Premature Convergence and Stagnation in the Discrete Differential Evolution Algorithm," *Applied Computational Intelligence and Soft Computing*, vol. 2023, no. 1, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[56] I.M. El-Desoky et al., "A Hybrid Genetic Algorithm for Job Shop Scheduling Problems," *International Journal of Advancement in Engineering, Technology and Computer Sciences*, vol. 3, no. 1, pp. 6-17, 2016. [Google Scholar]

[57] Warrick Pierce, and Monique Le Roux, "Statistics of Utility-Scales Power Generation in South Africa," *CSIR Energy Center*, pp. 1-122, 2023. [Google Scholar]

[58] Chnoor M. Rahman, and Tarik A. Rashid, "A New Evolutionary Algorithm: Learner Performance based Behavior Algorithm," *Egyptian Informatics Journal*, vol. 22, no. 2, pp. 213-223, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[59] Jaza Mahmood Abdullah, and Tarik Ahmed, "Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process," *IEEE Access*, vol. 7, pp. 43473-43486, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[60] Janez Brest, Mirjam Sepesy Maučec, and Borko Bošković, "The 100-Digit Challenge: Algorithm jDE100," *2019 IEEE Congress on Evolutionary Computation*, Wellington, New Zealand, pp. 19-26, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[61] Gulin Zeynep Oztas, and Sabri Erdem, "A Penalty-based Algorithm Proposal for Engineering Optimization Problems," *Neural Computing and Applications*, vol. 35, no. 10, pp. 7635-7658, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[62] Saeed Rafee Nekoo, José Ángel Acosta, and Anibal Ollero, "A Search Algorithm for Constrained Engineering Optimization and Tuning the Gains of Controllers," *Expert Systems with Applications*, vol. 206, pp. 1-27, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[63] Chalermjit Klansupar, and Surachai Chaitusaney, "Optimal Sizing of Grid-Scaled Battery with Consideration of Battery Installation and System Power-Generation Costs," *Energies*, vol. 15, no. 13, pp. 1-18, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[64] Jonas Engels, Bert Claessens, and Geert Deconinck, "Optimal Combination of Frequency Control and Peak Shaving with Battery Storage Systems," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 3270-3279, 2020. [CrossRef] [Google Scholar] [Publisher Link]