Research Article

A Master-Slave Salp Swarm Algorithm Optimizer for Hybrid Energy Storage System Control Strategy in Electric Vehicles

Fabian Cheruiyot and Davies Segera

Department of Electrical and Information Engineering, University of Nairobi, Nairobi 30197, Kenya

Correspondence should be addressed to Fabian Cheruiyot; cheruiyotfabian@gmail.com

Received 28 May 2022; Revised 23 July 2022; Accepted 20 August 2022; Published 14 September 2022

Academic Editor: Edith Osorio de la Rosa

Copyright © 2022 Fabian Cheruiyot and Davies Segera. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Pure electric vehicles provide an enticing ecofriendly alternative to traditional fossil fuel combustion engine locomotives. Batteries have primarily been used to store energy in electric vehicles; however, peak load demand and transient power leading to decreased battery lifespan have bred interest in hybrid energy storage systems in electric vehicles. Management of energy drawn from a hybrid energy storage system (HESS) in electric vehicles is a real-time multistage optimization problem aimed at minimizing energy consumption while aptly distributing energy drawn from the battery and capacitor to enhance the battery life cycle. This paper explores the feasibility of a master-slave salp swarm optimization algorithm (MSSSA) (metaheuristic algorithm) in a HESS control strategy for electric vehicles. Introducing a master-slave learning approach to the salp swarm algorithm (SSA) improves its performance by increasing its convergence rate while maintaining a balance between exploration and exploitation phases of the algorithm. A comparison of the MSSSA results with the SSA (salp swarm algorithm), DA (dynamic algorithm), WOA (whale optimization algorithm), MFO (moth flame optimization algorithm), GA (genetic algorithm), and PSO (particle swarm optimization algorithm) on benchmark test functions and dynamic program simulation of an electric vehicle’s HESS control strategy and shows preeminence of the MSSSA control strategy for HESS.

1. Introduction

Electric vehicles are undoubtedly the future of the automobile industry. Owing to their energy saving ability and lack of exhaust pollution, electric vehicles have high potential of replacing combustion fuel vehicles. However, pure electric vehicles are still plagued with limitations such as poor energy consumption, long charging periods, and short battery range. In interest, to extend the mileage and energy efficiency of EVs led to the incorporation of hybrid energy storage systems (HESS) composed of batteries and capacitors. HESS incorporates the advantages of both batteries and capacitors for an optimal balance between capacity, charge/discharge time, and life cycles. To maintain this balance, management and control systems for HESS have been explored in energy distribution based on the working conditions of an EV [1].

With emphasis on real time-energy consumption optimization, fuzzy control is at the core of these management and control systems applied by the industry. Recent developments in machine learning and the development of metaheuristic algorithms have seen scholars take on optimization algorithms research in management and control systems of HESS [2]. These algorithms are particularly promising in the engineering field as they are able to abstractly solve optimization problems, significantly reducing time and resource taken to solve them despite presenting variable performances.

Metaheuristic nature-inspired optimization algorithms have become a fervent field in extreme learning machine (ELM). ELM is an extremely fast training method for single hidden layer feed forward networks (SLFN), the most prevalent artificial neural network (ANN) [3]. As suggested in the name “nature-inspired,” these algorithms are inspired from natural phenomena with optimization evident at the core of the living organisms’ survival mechanisms. These algorithms include the following: genetic algorithm (GA), grey wolf optimizer (GWO), dragon-fly...
algorithm, whale optimization algorithm, and ant colony optimization (ACO) [4].

Over the years, attempts to improve ELM performance led to the development of metaheuristic algorithms. These techniques have become popular owing to their flexibility in solving diverse problems, their higher immunity to being trapped in a local minimum, and that they apply a gradient-free mechanism, hence more likely to converge at a solution faster [3]. Since they are nature-inspired, metaheuristic algorithms can optimally arrive at a solution without traversing the entire problem search space, therefore achieving both speed and reducing computation costs [5].

The salp swarm algorithm (SSA) proposed by Mirjalili et al. in 2017 mimics the behavior of a swarm of salps in nature [6]. Salps are aquatic creatures with a resemblance to the jellyfish’s transparent body, however, with a distinctive barrel body shape. In deep oceans, salps exist in swarms known as salp chains. The swarming nature of salps is evident as they form collaborative chains during foraging activities in the deep ocean. Swarming of salps is believed to aid in gaining kinetic energy as they pursue a moving food source.

In electrical engineering, practical optimization problems such HESS control involve determining the optimal parameters based on one or more linear/nonlinear equality/inequality constraints [7]. Implementation of stochastic optimization methods has become popular in the field. The salp swarm algorithm has been used in parameter identification for electrical engineering optimization problems and proven to outperform other methods in terms of accuracy while maintaining prediction stability. However, SSA has similar limitations to other metaheuristic algorithms such as slow convergence due to its low exploitation ability [8].

In the case of real-time optimization problems, fuzzy control has predominantly been adopted ensuing from the necessity of speed in these operations. This method compromises on performance for speed. Fuzzy control is also rigid as logic rules have to be preemptively defined for varying working conditions. Contrast to this, metaheuristic algorithms are generally slower. Nevertheless, these algorithms have been proven to have significant performance lead whilst being versatile and pliable to the nature and conditions of the problem.

This paper proposes Master-Slave Salp Swarm Optimizer for real-time optimization of HESS control strategy of pure EVs. In computing, the term master slave was adopted referring to the asymmetric control/communication models. In this model, devices or processes are grouped into one master and several slave groups. The master controls one or more processes or devices (known as slaves) and acts as a communication hub between the slaves [9]. Introducing a master-slave approach to the optimization algorithm is endeavored towards improving the ability to maintain balance between the exploration and exploitation phases of the SSA, thus improving its ability to skip local optimum points reaching the global optimum solution faster by eliminating trap solutions [10, 11]. By implementing an improved adaptive power distribution learning algorithm in a simulated environment, the theoretical minimum energy consumption from a management and control system for a HESS can be calculated [12].

2. Literature Review

The concept of design and implementation of electric vehicles dates back a hundred years ago. A major roadblock to commercial production and application of electric vehicles has been their range limitation [2]. To overcome this limitation, the development of hybrid vehicles began in the early 2000s. However, in the late 2000s, improvement in battery manufacturing technology has seen the concept of pure electric vehicles, augmented from experimental to a reality and future of the motor industry [13].

Pure electric vehicles existing today have a practical range of about 160 km to 600 km, which is a significant surge [14] but still insufficient to contend with combustion engine vehicle sufficiency. To achieve this, the application of HESS in EVs was explored [1]. HESS are composed of two or more types of energy storage devices/technologies that complement each other (typically batteries and capacitors) [15]. Their complementary nature enables them to outperform any conventional single component energy storage device such as batteries, fuel cells, and super capacitors [16]. Improving this range not only hinges to further enhancement of energy storage systems in electric vehicles but also the management and control of energy distribution for the best efficiency in different working conditions.

Composite power supply structures (HESS) are commonly classified under three categories. These include active, semiaactive, and passive [15]. Active composite power supply consists of two integrated DC/DC converters serially connected with batteries and capacitors. For passive HESS, a battery and ultracapacitor are directly connected in parallel and the voltage of the two synchronized in real time. The energy distribution between the two is not adjustable, thus constant for full capacity and high-power output conditions. Semiactive HESS is a compromise between performance and cost having only one DC/DC converter with a variety of control strategies being applicable. Semiactive structures are of two types: battery end load and capacitor end load.

With optimization in mind, a new battery/ultracapacitor HESS for electric vehicles was proposed [16, 17]. The system incorporates the use of a controlled energy pump in the form of a small DC/DC converter to keep the ultracapacitor’s voltage higher compared to the battery voltage in conventional urban drive conditions. For connections, a modulator takes the place of the DC/DC convertor, thus resolving the problem of large voltage fluxes emanating from the ultracapacitor power transmission. This system allows for energy control and optimization without affecting the performance of the drive motor [2].

2.1. HESS Energy Optimization. In an ideal situation, the input and output power of a system are equal; meaning, the change in acceleration of an electric vehicle directly reflects the change in power drawn from HESS [12]. Practically, lithium-ion batteries and ultracapacitors are used as electrical energy carriers with varying remaining stored
energy in an ever-changing operating environment. The trend in energy consumed over working time has a close relationship with the operating mode, ambient temperature, and switching losses associated. Energy management strategy research is focused on the optimization of EV energy consumption without fundamentally changing the base components and the framework of HESS [18]. The proposed optimization in HESS is based on calculation of the suboptimal current to control it and minimize the working current and battery fluctuations in EVs. HESS energy optimization has further been represented as three optimization problems [1, 12, 19]:

(i) Minimization of battery current fluctuation: high amount of fluctuations in the battery current magnitude causes an increase in the battery’s internal resistance. Consequently, this reduces the battery longevity

(ii) Minimization of energy loss in HESS due to ultracapacitor ESR that lowers the battery discharge time: as a result, extending the battery discharge phase depends on reducing energy loss

(iii) Dual-objective real-time minimization, for optimal performance: there is a need to minimize both battery current fluctuations and capacitor energy losses while meeting vehicle energy requirements in different working conditions

2.2. Fuzzy Control Strategy for HESS. HESS working has two states, charging and discharging. During the drive cycle, HESS devices are being alternately being discharged and charged [17], considering the regenerative braking which has been integrated in pure electric vehicles. Under different working conditions, the power demand is different and so is the control strategy formulation required [1]. The power demand can be split into two: peak power and average power. Since instantaneous high current discharge must be prevented, ultracapacitors are responsible for the peak power and average power drawn from lithium-ion batteries [15]. Ultracapacitors are also responsible for ensuring the maximum possible energy is recovered during braking.

While the power demand is greater than 0, the HESS is discharging, and the control strategy follows that both the UC and battery are discharged at the same rate; however, the lithium-ion battery must also supply power to the ultracapacitor ensuring that it always has enough energy to meet peak power demand fluctuations [15, 16].

(i) While the power demand is very small and the power stored in the battery is or moderate or higher, the battery discharges alone or separately where as if the power demand is small and the ultracapacitor stored energy is higher or moderate, both discharge together; otherwise, the battery discharges separately

(ii) While the power demand is moderate and both the battery and ultracapacitor stored energy are higher, both discharge together; otherwise, the lithium-ion battery discharges alone. If the battery energy is low, both the capacitor and battery discharge together

(iii) While the power demand is high and the battery stored energy and the capacitor energy are moderate or higher, both will discharge together. Otherwise, the capacitor will discharge separately

While the power demand is less than 0, the HESS is in charging mode, and the control strategy is based on the residual capacities of the battery and ultracapacitor. The capacitor is charged as frequently as the battery, and the battery not only takes the input power required for the vehicle operation but also the input power of the ultracapacitor to ensure that enough energy is available to cope with large power fluctuations in the drive cycle. Typically, both are charged together; however, in conditions where a maximum amount of energy needs to be stored over a short period, i.e., in the case of regenerative braking, the ultracapacitor is charged separately [19].

Management of flow energy from and to different sources separately was achieved by a combination of fuzzy control and shape control. Studies focused on an online predictive control strategy proposed a dual-loop online intelligent planning (DOIP) method for speed forecasting and energy control [13]. Optimal control behavior was thus determined by being able to learn the speed and acceleration of an electric vehicle under different conditions. Fuzzy control in electric vehicle HESS is used to set different control variables that control different components thus improving the energy efficiency by establishing a range of fuzzy logic rules [18]. The rules can be grouped into a set of two: fuzzy control rules for the output power consumed in driving and fuzzy logic rules for power recovery when braking [12, 18].

Fuzzy controllers have been essential in HESS control strategies as they are able to cover a number of constraints by incorporating multiple fuzzy logic rules. Optimization of these controllers, however, becomes complex considering the sheer number of parameters composed of boundaries and variables that constitute the logic rules [12]. This increases the time required for optimization significantly lending the application of metaheuristic optimization algorithms impractical or otherwise expensive considering the required compute power to speed up these computations to a fraction of a second.

2.3. Algorithm Optimization. Applying algorithm optimization requires the problem to be modelled mathematically [7]. Optimization is the process for training the model iteratively to result in a minimum and maximum function evaluation. Optimization algorithms are split into two phases: exploration phase and exploitation phase [11]. In the exploration phase, the solutions are gravitated towards the different regions of the problem search space by sudden changes. This is aimed at identifying promising areas in the search space and to avoid local solutions. In the exploitation phase, the goal is to improve the accuracy by gradually changing
solutions [20]. The results of every iteration are compared by changing the hyperparameters until an optimum result is achieved. Maxima is the largest value of a function and minima the smallest value within a given range. A global maxima and minima are the maximum and minimum values of a function, respectively, in the entire domain of a given function, where as local maxima and minima are given for a function within a particular range. There can only be one global maxima and global minima but more than one local maximum and minima.

Gradient descent is an optimization algorithm that finds out the local minima of a differentiable function. It minimizes a given function based on the learning rate. Learning rate is a hyperparameter determining the step size at each iteration while moving towards the minima of a function. Gradient descent has high time complexity, thus the need for implementation of stochastic gradient descent. Stochastic stands for probabilistic; therefore, random points from a population are selected, reducing the time complexity and allowing for faster convergence [20]. Metaheuristic algorithms such as the SSA discussed in this paper fall under this classification.

2.4. Salp Swarm Algorithm. The salp swarm algorithm (SSA) proposed by Mirjalili et al. in 2017 mimics the behavior of a swarm of salps in nature. Salps are aquatic creatures with a resemblance to the jellyfish’s transparent body, however, are barrel shaped. In deep oceans, salps exist in swarms known as salp chains as illustrated in Figure 1. The swarming nature of salps is evident as they form collaborative chains during foraging activities in the deep ocean. Swarming of salps is believed to aid in gaining kinetic energy as they pursue a moving food source [6].

In mathematical modelling, the salp chain is divided into two: leader and the rest are followers. The position of a salp is defined in a search space of dimension $d$, where $d$ is defined as the number of variables for a given problem. The swarms’ target is a food source expressed as $F$. The leader attacks in the direction of $F$ and the followers update their position towards the rest of the salps or the leader either directly or indirectly. A salp chain, $X$, in dimension $d$ consists of $N$ number of agents and can thus be expressed in a matrix form of $N \times d$

$$X_i = \begin{bmatrix} x^1_1 & x^1_2 & \cdots & x^1_d \\ x^2_1 & x^2_2 & \cdots & x^2_d \\ \vdots & \vdots & \ddots & \vdots \\ x^n_1 & x^n_2 & \cdots & x^n_d \end{bmatrix}. \quad (1)$$

The leader updates its position in the $j$-th dimension, $X^j_l$, according to the equation below:

$$X^j_l = \begin{cases} F_j + c_1 \left( (ub_j + lb_j)(c_2 + lb_j) \right) & \text{if } C_3 \geq 0.5 \\ F_j - c_1 \left( (ub_j - lb_j)(c_2 + lb_j) \right) & \text{if } C_3 < 0.5 \end{cases}, \quad (2)$$

where $F_j$ is the position vector of the food source, $ub_j$ is the superior limit of the $j$-th dimension, $lb_j$ is the inferior limit of the $j$-th dimension, and $c_1$ and $c_2$ are random values in the range $[0,1]$. The main parameter of the algorithm is $c_1$, expressed as

$$c_1 = 2e^{-\left( \frac{I}{I_{\text{max}}} \right)^2}, \quad (3)$$

with $I$ being the number of the current iteration, and $I_{\text{max}}$ is the maximum number of iterations. The followers’ position is updated utilizing Newton’s laws of motion:

$$x^j_i = \frac{1}{2}at^2 + v_0t. \quad (4)$$

With $a = \frac{v_{\text{max}}}{v_0}$ where $v_0$ is initial speed and $v$ is expressed as $v = x - x_0/t$ considering $v_0 = 0$ and the discrepancy between iterations to be 1, the above expression can be written as

$$x^j_i = \frac{x^i_j + x^{j-1}_i}{2}, \quad (5)$$

where $i \geq 2$. From these equations, a salp chain is simulated.

The SSA pseudocode, Algorithm 1 starts by initiating multiple salps with random positions and calculates the fitness of each to find the one with the highest fitness and assigns it to $F$. The leader will then be iteratively followed by the chain of salps using equation (5); meanwhile, $c_1$ is updated by equation (3) for each iteration, and equation (2) updates the leader’s position till the last iteration [6].

3. Methodology
3.1. Master-Slave Salp Swarm Algorithm. This paper proposes the master-slave salp swarm algorithm, Algorithm 2 following the SSA nature; however, the salp population is split into a master and slave population based on fitness such that the top half forms the master population and the bottom half, the slave population. Each slave salp learns from a salp in the master population [9]. The selection of these
The novelty of master-slave learning approach improves the exploitation phase of the SSA thus improving the convergence relative to the number of iterations. This is presumed to improve the relative performance of the SSA, however, runs the risk of limiting the algorithms exploration phase while still slightly increasing its complexity in terms of number of operations. A flowchart of the MSSSA is illustrated in Figure 2.

With coefficients \( r_1 \) and \( c_1 \) in the learning equation, \( L_q \) provides a random generally decreasing learning rate which is aimed at maintaining a balance between the exploration and exploitation phase of MS-SSA. These two features improve the algorithms performance over the others, especially where the luxury of high number of iterations cannot be afforded.

### 3.2. HESS Modeling

To test the viability of the MSSSA in the real-time optimization of energy distribution for HEES in electric vehicles, a basic model of the system is illustrated in Figure 3.

The hybrid energy storage system consists of a lithium-ion battery module in parallel with an ultracapacitor module [16]. Lithium-ion batteries have high storage capacity, hence favorable in powering electric vehicles today. However, lithium-ion batteries degrade with usage over time, and this is accelerated by fast discharging or charging over short periods of time [15]. Drawing large amounts of current may lead to overheating and thus the need for current

---

**Algorithm 1: SSA Algorithm pseudocode.**

```
Initialize the salp population \( x_i \) \((i=1, 2, ..., n)\) considering \( u_b \) and \( l_b \)
while (end condition is not satisfied)
Calculate the fitness of each search agent (salp)
\( F \) = the best search agent
update \( c_1 \) by Eq. (iii)
for each salp \( x_i \)
  if \((i=1)\)
    Update the position of the leading salp by Eq. (ii)
  else
    Update the position of the follower salp by Eq. (v)
  end
end
Amend the salps based on the upper and lower bounds of variables
return \( F \)
```

**Algorithm 2: MSSSA pseudocode.**

```
Initialize the salp population \( x_i \) \((i=1, 2, ..., n)\) considering \( u_b \) and \( l_b \)
while (end condition is not satisfied)
Calculate the fitness of each search agent (salp)
Sort the salps in ascending order of fitness and make the bottom half the slave population.
The slave population learn from the master population by equations (vi) and (vii)
\( F \) = the best search agent
update \( c_1 \) by Eq. (iii)
for each salp \( x_i \)
  if \((i=1)\)
    Update the position of the leading salp by Eq. (ii)
  else
    Update the position of the follower salp by Eq. (v)
  end
end
Amend the salps based on the upper and lower bounds of variables
return \( F \)
```
regulation. Ultracapacitors on the other hand have limited capacity but have the advantage of having very fast charging and discharge time without degrading. The incorporation of HESS in EVs is aimed at seizing the advantages of both to supply sufficient power required in a driving cycle while minimizing battery degradation [1]. Drawn from this knowledge, a suitable control strategy is modeled as in Figure 4 below.

3.2.1. Battery Model. Typically, EV battery module consists of a number of cells connected in series and parallel. The result is a high voltage, high-capacity module ranging from 200 V to 500 V and 20 kWh to 100 kWh. A simple model adapted consists of the battery as a voltage source and its internal resistance serially connected [21]. The voltage output depends on the state of charge as illustrated in (Figure 5).

\[
\text{SOC}_{\text{bat}} = \frac{Q_{\text{bat}}}{Q_n} = \frac{Q_{\text{bat}}}{Q_{\text{initial}}} \times \text{SOC}_{\text{bat initial}} \quad (8)
\]

where \( Q_{\text{bat}} \) is the current amount of energy stored in the battery module, and \( Q_n \) is the nominal capacity of the battery module. The battery voltage \( V_{\text{bat}} \) at 90% to 15% is approximated with the relation

\[
V_{\text{bat}} = V_{\text{bat}} - (\Delta \text{SOC}_{\text{bat}} \times K), \quad (9)
\]

where \( K \) is a constant of approximate value 0.0007.

Current drawn (\( I_{\text{bat}} \)) is calculated where a given load \( P_{\text{bat}} \) is applied to the terminals such that the total power drawn from the battery is expressed by

\[
V_{\text{bat}} \times I_{\text{bat}} = I_{\text{bat}}^2 \times R_{\text{bat}} + P_{\text{bat}}, \quad (10)
\]

where \( R_{\text{bat}} \) is the battery’s internal resistance.

Solving for \( I_{\text{bat}} \) from equation (10) results in a quadratic equation expressed as

\[
I_{\text{bat}} = \frac{V_{\text{bat}} \pm \sqrt{V_{\text{bat}}^2 - 4R_{\text{bat}} - P_{\text{bat}}}}{2R_{\text{bat}}}. \quad (11)
\]

3.2.2. Ultracapacitor Model. Ultracapacitors have much higher capacitance (about 400 F in EV HESS) compared to typical capacitors with low voltage limits at about 3 V per capacitor [22]. They have the advantage of storing significantly large amounts of energy per unit volume compared to electrolytic capacitors with higher charge and discharge power. They have very long-life cycles; hence, life decay can be neglected in the implementation of UCs (ultracapacitors) to store charge in EVs [18].

The voltage at the ultracapacitor terminals varies linearly with its state of charge. This voltage can be calculated from the energy stored in the UC and the capacitance by

\[
V_{\text{uc}} = \frac{0.5 \times Q_{\text{uc}}}{C_{\text{uc}}} + \sqrt{\left(\frac{0.25 \times Q_{\text{uc}}}{C_{\text{uc}}}\right)^2 - P_{\text{uc}} \times R_{\text{uc}}}, \quad (12)
\]

where \( Q_{\text{uc}} \) is energy stored by the UC, \( C_{\text{uc}} \) is the capacitance of the UC module, \( P_{\text{uc}} \) is the capacitor load, and \( R_{\text{uc}} \) is the UC modules internal Resistance.

Given that UCs are applied where large power is required over a short period of time, the current drawn can be approximated neglecting the internal resistance of the module (equation (5)).

\[
I_{\text{uc}} = \frac{P_{\text{uc}}}{V_{\text{uc}}}. \quad (13)
\]

State of charge of the UC is calculated as

\[
\text{SOC}_{\text{uc}} = \frac{Q_{\text{uc}}}{Q_{\text{uc}}(n)} = \frac{Q_{\text{uc}}}{Q_{\text{uc}}(\text{init})} \times \text{SOC}_{\text{uc}}(\text{init}). \quad (14)
\]
3.3. Objective Modelling. Control and optimization of HESS in EVs are achieved by a controller whose functions include energy demand distribution between the UC and battery modules, charge, and discharge rate regulation to enhance the battery’s life cycle and minimization of energy loss in the system [1].

During vehicle operations, the power required is determined from the vehicles’ mass, acceleration, drag, and losses in the mechanical system driving the vehicle [23]. Aerodynamic drag is one of the major inhibitions to acceleration through air and is expressed as

$$\text{Drag} = C_d \cdot \frac{\rho \cdot v^2}{2} \cdot A,$$  \hfill (15)

where $C_d$ is the drag coefficient, approximately, $\rho$ is the air density, $v$ is the vehicles velocity, and $A$ is the reference area.

During acceleration, the force required is given by

$$F = m \cdot a,$$  \hfill (16)

where $m$ is the mass of the vehicle, and $a$ is its desired acceleration.

With efficiency of the system as $\eta$, the energy demand can thus be calculated by multiplying the sum of $F$ and drag with velocity and time. Over a period of 1 sec, $E_{\text{demand}}$ is expressed by

$$E_{\text{demand}} = \left( (m \cdot a(t)) + \left( C_d \cdot \frac{\rho \cdot v(t)^2}{2} \cdot A \right) \right) \cdot V \cdot \eta.$$  \hfill (17)

This energy demand has to be met by the sum of the battery power and UC power. Taking a period time of 1 sec, the energy demand ($E_{\text{demand}}$) is equal to the sum of $P_{\text{bat}}$ and $P_{\text{uc}}$, ignoring the system’s losses.

$$E_{\text{demand}} = P_{\text{bat}} + P_{\text{uc}}.$$  \hfill (18)

Outputs of the battery and ultracapacitor depend on the SOC of the battery/UC and the power demand such that the current limits of the battery are not exceeded [12]. To
describe this, the energy distribution factors $K_{bat}$ and $K_{uc}$ were proposed as follows [12]:

$$P_{bat} = K_{bat} \times E_{demand},$$

$$P_{uc} = K_{uc} \times E_{demand},$$

$$K_{uc} = 1 - K_{bat}.$$  \hspace{1cm} (19)

The mathematical model of the objective function is described as

$$\min y = f(x).$$ \hspace{1cm} (20)

With energy consumption per unit distance as the evaluation standard, it is expressed as

$$f(x) = \text{fitness} = \frac{\text{energy}}{\text{distance}}.$$ \hspace{1cm} (21)
| Function                        | Best score | SSA       | WOA       | GA         | MFO       | PSO       | DA         |
|--------------------------------|------------|-----------|-----------|------------|-----------|-----------|------------|
| Sphere f1                       | 4.98E-09   | 4.89E-09  | 1.23E-189 | 1.80E+01   | 9.48E-06  | 4.30E-02  | 1.46E-02   |
| Best score                      | 4.77E-06   | 4.26E-06  | 1.93E-123 | 1.40E+01   | 3.50E-20  | 1.24E+01  | 4.08E-01   |
| Worst score                     | 7.54E-06   | 1.25E-05  | 2.20E-114 | 1.90E+01   | 2.55E-18  | 7.39E-01  | 3.92E+00   |
| Mean                            | 6.10E-06   | 6.95E-06  | 4.42E-115 | 1.64E+01   | 4.57E-19  | 5.03E-01  | 1.92E+00   |
| Standard deviation              | 9.15E-07   | 2.26E-06  | 9.18E-115 | 1.84E+00   | 7.66E-19  | 1.43E-01  | 1.33E+00   |
| Schwefel's problem 2.22 f2      | 9.93E-10   | 5.90E-10  | 2.40E-09  | 2.20E+01   | 2.24E-13  | 3.65E+01  | 3.40E-03   |
| Best score                      | 9.92E-06   | 8.94E-06  | 0.00E+00  | 1.40E+01   | 1.61E-10  | 1.06E+00  | 1.97E-01   |
| Worst score                     | 1.92E-05   | 1.93E-05  | 3.00E+01  | 2.00E+01   | 1.48E-01  | 1.85E+00  | 7.95E+00   |
| Mean                            | 1.51E-05   | 1.32E-05  | 3.06E+00  | 1.57E+01   | 1.48E-02  | 1.34E+00  | 3.58E+00   |
| Standard deviation              | 3.14E-06   | 2.97E-06  | 9.46E+00  | 1.89E+00   | 4.67E-02  | 2.68E-01  | 2.36E+00   |
| Schwefel's problem 2.21 f3      | 3.94E+00   | 3.45E+00  | 5.72E+00  | 0.00E+00   | 4.39E+00  | 9.88E+01  | 7.57E+00   |
| Best score                      | 1.49E+03   | 1.43E+03  | 6.59E+00  | 0.00E+00   | 9.00E+04  | 4.82E+02  | 8.41E+02   |
| Worst score                     | 1.78E+02   | 2.01E+02  | 6.01E+00  | 0.00E+00   | 9.61E+03  | 2.37E+02  | 1.92E+02   |
| Mean                            | 4.65E+02   | 4.52E+02  | 2.60E+01  | 0.00E+00   | 2.83E+04  | 1.34E+02  | 2.76E+02   |
| Standard deviation              | 3.67E-10   | 2.96E-10  | 1.12E-04  | 4.35E+01   | 0.00E+00  | 3.98E-02  | 8.00E-04   |
| Schwefel's problem 2.211 f4     | 8.84E-10   | 8.77E-10  | 9.30E-06  | 5.55E+00   | 7.61E-31  | 3.51E-01  | 7.83E+00   |
| Best score                      | 5.97E-10   | 5.92E-03  | 5.20E-05  | 5.09E+01   | 1.18E-31  | 1.46E-01  | 8.79E-01   |
| Worst score                     | 1.69E-10   | 1.84E-10  | 3.50E-05  | 4.12E+00   | 2.34E-31  | 8.25E-02  | 2.45E+00   |
| Mean                            | 1.70E-03   | 1.10E-03  | 0.00E+00  | 2.10E+01   | 1.06E-03  | 6.20E-01  | 2.60E-03   |
| Standard deviation              | 2.22E-03   | 3.27E-03  | 1.13E-03  | 1.54E+00   | 1.85E-03  | 5.47E-01  | 6.07E-03   |
| Step function f6                | -3.02E+03  | -3.42E+03 | -4.19E+03 | -9.26E+00  | -3.71E+03 | -7.73E+03 | -3.37E+00  |
| Best score                      | -2.27E+03  | -2.61E+03 | -2.50E+03 | -5.05E+00  | -2.76E+03 | -5.03E+03 | -2.13E+00  |
| Worst score                     | -2.72E+03  | -2.97E+03 | -3.39E+03 | -6.65E+00  | -3.36E+03 | -6.51E+03 | -2.70E+00  |
| Mean                            | 2.29E+02   | 2.27E+02  | 5.39E+02  | 1.56E+00   | 3.23E+02  | 8.66E+02  | 3.40E-01   |
| Quadratic function with noise f7 | -3.02E+03  | -3.42E+03 | -4.19E+03 | -9.26E+00  | -3.71E+03 | -7.73E+03 | -3.37E+00  |
| Generalized Schwefel's problem 2.26 f8 | -2.27E+03  | -2.61E+03 | -2.50E+03 | -5.05E+00  | -2.76E+03 | -5.03E+03 | -2.13E+00  |
| Function Description                  | f9   | f10  | f11  | f12  | f13  | f14  | f15  | f16  |
|--------------------------------------|------|------|------|------|------|------|------|------|
| Generalized Rastrigin's function     |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-04 | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Generalized Griewank function        |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Generalized penalized function       |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Generalized penalized function       |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Shekel's foxhole function            |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Shekel's foxhole function            |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Six-hump camel back function         |      |      |      |      |      |      |      |      |
| Mean                                 | 1.33E-01 | 4.90E-01 | 3.11E-02 | 3.11E-02 | 1.96E-01 | 9.96E-01 | 8.07E-4  | -1.03E-00 |
| Standard deviation                    | 1.37E-01 | 6.17E-02 | 3.11E-01 | 3.11E-01 | 1.00E-01 | 1.99E+01 | 7.93E-04 | -1.03E+00 |
| Worst score                          | 7.19E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
| Best score                           | 3.98E+00 | 7.57E-06 | 9.85E-02 | 9.33E-02 | 1.28E+01 | 9.98E-01 | 1.02E+00 | -1.03E+00 |
Table 1: Continued.

| Function                  | MSSSA     | SSA       | WOA       | GA        | MFO       | PSO       | DA        |
|---------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Branin function           |           |           |           |           |           |           |           |
| f7                        |           |           |           |           |           |           |           |
| Best score                | NA        | NA        | 3.98E-01  | 3.58E+01  | 3.98E-01  | 3.98E-01  | 3.98E-01  |
| Worst score               | NA        | NA        | 3.98E-01  | 5.56E+01  | 3.98E-01  | 3.98E-01  | 3.98E-01  |
| Mean                      | NA        | NA        | 3.98E-01  | 4.53E+01  | 3.98E-01  | 3.98E-01  | 3.98E-01  |
| Standard deviation        | NA        | NA        | 0.00E+00  | 8.11E+00  | 0.00E+00  | 0.00E+00  | 0.00E+00  |
| Goldstein-Price function  |           |           |           |           |           |           |           |
| f8                        |           |           |           |           |           |           |           |
| Best score                | 3.00E+00  | 3.00E+00  | 3.00E+00  | 6.00E+02  | 3.00E+00  | 3.00E+00  | 3.00E+00  |
| Worst score               | 2.35E+01  | 3.00E+00  | 3.00E+00  | 2.86E+04  | 3.00E+00  | 3.00E+00  | 3.00E+00  |
| Mean                      | 8.05E+00  | 3.00E+00  | 3.00E+00  | 9.42E+03  | 3.00E+00  | 3.00E+00  | 3.00E+00  |
| Standard deviation        | 7.94E+00  | 0.00E+00  | 0.00E+00  | 1.33E+04  | 0.00E+00  | 0.00E+00  | 0.00E+00  |
| Hartman’s family          |           |           |           |           |           |           |           |
| f9                        |           |           |           |           |           |           |           |
| Best score                | -3.86E+00 | -3.86E+00 | -3.86E+00 | NA        | -3.86E+00 | -3.86E+00 | -3.86E+00 |
| Worst score               | -3.29E+00 | -3.86E+00 | -3.86E+00 | NA        | -3.86E+00 | -3.86E+00 | -3.86E+00 |
| Mean                      | -3.61E+00 | -3.86E+00 | -3.86E+00 | NA        | -3.86E+00 | -3.86E+00 | -3.86E+00 |
| Standard deviation        | 1.82E-01  | 0.00E+00  | 6.73E-04  | NA        | 0.00E+00  | 0.00E+00  | 0.00E+00  |
| Hartman’s family          |           |           |           |           |           |           |           |
| f20                       |           |           |           |           |           |           |           |
| Best score                | -3.32E+00 | -3.32E+00 | -3.32E+00 | NA        | -3.20E+00 | -3.32E+00 | -3.32E+00 |
| Worst score               | -2.28E+00 | -3.20E+00 | -3.15E+00 | NA        | -3.14E+00 | -3.20E+00 | -3.05E+00 |
| Mean                      | -3.02E+00 | -3.21E+00 | -3.27E+00 | NA        | -3.19E+00 | -3.27E+00 | -3.23E+00 |
| Standard deviation        | 3.88E-01  | 3.77E-02  | 7.06E-02  | NA        | 2.65E-02  | 6.14E-02  | 1.01E-01  |
| Shäkel’s family           |           |           |           |           |           |           |           |
| f21                       |           |           |           |           |           |           |           |
| Best score                | -1.02E+01 | -1.02E+01 | -1.02E+01 | NA        | -1.02E+01 | -1.02E+01 | -1.02E+01 |
| Worst score               | -5.10E+00 | -2.63E+00 | -1.02E+01 | NA        | -2.68E+00 | -5.06E+00 | -5.10E+00 |
| Mean                      | -9.14E+00 | -9.40E+00 | -1.02E+01 | NA        | -6.87E+00 | -8.13E+00 | -9.65E+00 |
| Standard deviation        | 2.13E+00  | 2.38E+00  | 1.71E-04  | NA        | 2.92E+00  | 2.61E+00  | 1.60E+00  |
| Shäkel’s family           |           |           |           |           |           |           |           |
| f22                       |           |           |           |           |           |           |           |
| Best score                | -1.04E+01 | -1.04E+01 | -1.04E+01 | NA        | -1.04E+01 | -1.04E+01 | -1.04E+01 |
| Worst score               | -5.09E+00 | -5.13E+00 | -5.09E+00 | NA        | -2.77E+00 | -5.13E+00 | -2.77E+00 |
| Mean                      | -9.87E+00 | -9.88E+00 | -9.34E+00 | NA        | -9.11E+00 | -9.88E+00 | -9.11E+00 |
| Standard deviation        | 1.68E+00  | 1.67E+00  | 2.24E+00  | NA        | 2.78E+00  | 1.67E+00  | 2.78E+00  |
| Shäkel’s family           |           |           |           |           |           |           |           |
| f23                       |           |           |           |           |           |           |           |
| Best score                | -1.05E+01 | -1.05E+01 | -1.05E+01 | NA        | -1.05E+01 | -1.05E+01 | -1.05E+01 |
| Worst score               | -1.05E+01 | -1.05E+01 | -1.05E+01 | NA        | -2.87E+00 | -5.13E+00 | -2.87E+00 |
| Mean                      | -1.05E+01 | -1.05E+01 | -1.05E+01 | NA        | -7.80E+00 | -1.00E+01 | -7.80E+00 |
| Standard deviation        | 0.00E+00  | 0.00E+00  | 2.80E-04  | NA        | 3.59E+00  | 1.71E+00  | 0.00E+00  |
Table 2: MSSSA Wilcoxon rank sum test results.

|      | SSA  | WOA  | GA   | MFO  | PSO  | DA   |
|------|------|------|------|------|------|------|
| F1   |      |      |      |      |      |      |
| p    | 0.273| 0.0002| 0.0001| 0.0002| 0.0002| 0.0002|
| h    | 0    | 1    | 1    | 1    | 1    | 1    |
| zval | 1.0961| 3.7418| -3.7979| -3.7433| -3.7418| -3.7418|
| Rank sum | 120 | 155 | 55 | 55 | 55 | 55 |
| F2   |      |      |      |      |      |      |
| p    | 0.3847| 0.0002| 0.0002| 0.0002| 0.0002| 0.0002|
| h    | 1    | 1    | 1    | 1    | 1    | 1    |
| zval | -2.0693| 3.7418| -3.7503| 3.7418| -3.7418| -3.7418|
| Rank sum | 82 | 155 | 55 | 155 | 55 | 55 |
| F3   |      |      |      |      |      |      |
| p    | 0.6232| 1.83E-04| 1.61E-04| 1.61E-04| 0.0058| 1.83E-04|
| h    | 0    | 1    | 1    | 1    | 1    | 1    |
| zval | -0.4914| -3.7418| -3.7732| -3.7732| 2.7591| -3.7418|
| Rank sum | 98 | 55 | 55 | 145 | 55 | 55 |
| F4   |      |      |      |      |      |      |
| p    | 0.241| 0.0256| 1.73E-04| 0.0028| 1.77E-04| 1.82E-04|
| h    | 0    | 1    | 1    | 1    | 1    | 1    |
| zval | 1.1726| -2.2317| -3.756| 2.987| -3.7503| -3.7433|
| Rank sum | 121 | 75 | 55 | 145 | 55 | 55 |
| F5   |      |      |      |      |      |      |
| p    | 0.4274| 0.0538| NA   | 0.8501| 0.0101| 6.39E-05|
| h    | 0    | 0    | NA   | 0    | 1    | 1    |
| zval | 0.7937| 1.9283| NA   | -0.1891| -2.5711| 3.9981|
| Rank sum | 116 | 131 | NA | 152 | 70 | 155 |
| F6   |      |      |      |      |      |      |
| p    | 9.70E-01| 1.83E-04| 1.74E-04| 1.82E-04| 1.81E-04| 1.03E-04|
| h    | 0    | 1    | 1    | 1    | 1    | 1    |
| zval | 3.78E-02| -3.74E+00| -3.75E+00| 3.74E+00| -3.74E+00| -1.48E+00|
| Rank sum | 106 | 55 | 55 | 155 | 55 | 55 |
| F7   |      |      |      |      |      |      |
| p    | 1    | 0.0011| 1.82E-04| 0.014| 1.83E-04| 1    |
| h    | 0    | 1    | 1    | 1    | 1    | 0    |
| zval | 0    | 3.2542| -3.7433| 2.4568| -3.7418| 0    |
| Rank sum | 105 | 148.5 | 55 | 138 | 55 | 105 |
| F8   |      |      |      |      |      |      |
| p    | 4.09E-02| 2.57E-02| 1.75E-04| 1.30E-03| 1.82E-04| 1.82E-04|
| h    | 1    | 1    | 1    | 1    | 1    | 1    |
| zval | 2.04E+00| 2.23E+00| -3.75E+00| 3.22E+00| 3.74E+00| -3.74E+00|
| Rank sum | 132.5 | 135 | 55 | 148 | 155 | 55 |
| F9   |      |      |      |      |      |      |
| p    | 8.80E-01| NA   | 6.29E-05| 4.03E-01| 1.81E-04| 2.56E-02|
| h    | 0    | NA   | 1    | 0    | 1    | 1    |
| zval | -1.52E-01| NA   | -4.00E+00| -8.36E-01| -3.74E+00| -2.23E+00|
| Rank sum | 102.5 | NA | 55 | 93 | 55 | 75 |
|       | SSA    | WOA    | GA     | MFO    | PSO    | DA     |
|-------|--------|--------|--------|--------|--------|--------|
| **F10** |        |        |        |        |        |        |
| *p*   | 5.96E-01 | 1.40E-04 | NA     | 6.34E-05 | 1.82E-04 | 2.80E-03 |
| *h*   | 0      | 1      | NA     | 1      | 1      | 1      |
| zval  | -5.30E-01 | 3.81E+00 | NA     | 4.00E+00 | -3.74E+00 | -2.99E+00 |
| Rank sum | 97.5  | 155    | NA     | 155    | 55     | 65     |
| **F11** |        |        |        |        |        |        |
| *p*   | 4.73E-01 | 2.40E-01 | 6.39E-05 | 2.57E-02 | 1.83E-04 | 2.11E-02 |
| *h*   | 0      | 0      | 1      | 1      | 1      | 1      |
| zval  | -7.18E-01 | 2.95E+00 | -4.00E+00 | 2.23E+00 | 3.74E+00 | -2.31E+00 |
| Rank sum | 95     | 84     | 55     | 135    | 155    | 74     |
| **F12** |        |        |        |        |        |        |
| *p*   | 3.08E-01 | 2.80E-03 | 6.39E-05 | 2.54E-02 | 2.80E-03 | 1.00E-03 |
| *h*   | 1      | 1      | 1      | 1      | 1      | 1      |
| zval  | -1.06E+00 | -2.99E+00 | -4.00E+00 | 2.24E+00 | -2.99E+00 | -3.29E+00 |
| Rank sum | 87     | 65     | 55     | 135    | 65     | 61     |
| **F13** |        |        |        |        |        |        |
| *p*   | 3.93E-01 | 7.40E-01 | 4.88E-05 | 1.29E-01 | 8.15E-05 | 1.24E-04 |
| *h*   | 0      | 0      | 1      | 0      | 1      | 1      |
| zval  | -8.54E-01 | 3.32E-01 | -4.06E+00 | 1.52E+00 | -3.94E+00 | -3.84E+00 |
| Rank sum | 113    | 132    | 66     | 143    | 66     | 66     |
| **F14** |        |        |        |        |        |        |
| *p*   | NA     | 3.68E-01 | NA     | 3.68E-01 | 9.33E-02 | 1.47E-02 |
| *h*   | 0      | 0      | NA     | 0      | 0      | 1      |
| zval  | NA     | -9.00E-01 | NA     | -9.00E-01 | -1.68E+00 | -2.44E+00 |
| Rank sum | 105    | 100    | NA     | 100    | 95     | 80     |
| **F15** |        |        |        |        |        |        |
| *p*   | 9.30E-03 | 2.62E-02 | 1.22E-04 | 4.34E-02 | 4.59E-01 | 1.00E-03 |
| *h*   | 1      | 1      | 1      | 1      | 0      | 1      |
| zval  | -2.60E+00 | 2.22E+00 | -3.84E+00 | -2.02E+00 | -7.40E-01 | -3.29E+00 |
| Rank sum | 86.5   | 153    | 66     | 92     | 110    | 74     |
| **F16** |        |        |        |        |        |        |
| *p*   | NA     | NA     | 5.59E-05 | NA     | NA     | NA     |
| *h*   | 0      | 0      | 1      | 0      | 1      | 2      |
| zval  | NA     | NA     | -4.03E+00 | NA     | NA     | NA     |
| Rank sum | 105    | 105    | 55     | 105    | 106    | 107    |
| **F17** |        |        |        |        |        |        |
| *p*   | NA     | NA     | NA     | NA     | NA     | NA     |
| *h*   | NA     | NA     | NA     | NA     | NA     | NA     |
| zval  | NA     | NA     | NA     | NA     | NA     | NA     |
| Rank sum | NA     | NA     | NA     | NA     | NA     | NA     |
| **F18** |        |        |        |        |        |        |
| *p*   | 1.49E-02 | 1.49E-02 | 1.52E-04 | 1.49E-02 | 1.49E-02 | 1.49E-02 |
| *h*   | 1      | 1      | 1      | 1      | 1      | 1      |
| zval  | 2.43E+00 | 2.43E+00 | -3.79E+00 | 2.43E+00 | 2.43E+00 | 2.43E+00 |
| Rank sum | 130    | 130    | 55     | 130    | 130    | 130    |
Consumption of energy in HESS is composed pf various components including motor losses, line losses, DC/DC converter loss, capacitor, and battery losses. The main losses considered for optimization are the capacitor and battery losses, i.e.,

\[
\text{Energy} = P_{\text{bat}} + P_{\text{uc}} + E_{\text{bat}}^l + E_{\text{uc}}^l,
\]

\[
E_{\text{bat}}^l = I_{\text{bat}}^2(t) \cdot R_{\text{bat}},
\]

\[
E_{\text{uc}}^l = I_{\text{uc}}^2(t) \cdot R_{\text{uc}}.
\]

\[P_{\text{bat}} \text{ and } P_{\text{uc}} \text{ represent the output power of the battery and the ultracapacitor, respectively, with } E_{\text{bat}}^l \text{ and } E_{\text{uc}}^l \text{ representing the losses due to the battery and ultracapacitor, respectively. These are calculated from the working currents of the HESS components as shown (6) above.}

The algorithm optimization function can be concisely expressed as

\[
f(x) = \text{fitness} = \frac{E_{\text{bat}} + E_{\text{uc}}}{\text{distance}},
\]

where $E_{\text{bat}}$ is the energy drawn from the battery with battery loss considered, and $E_{\text{uc}}$ is the energy drawn from the UC with UC loss considered.

With the assumption that when SOC of the battery and UC are both high, i.e., close to 1, SOC has little to no impact on the energy distribution between the battery and the UC. Consequently, the only constraint is the current limits, and the optimization objective constraints are

\[
\left\{ \begin{array}{c}
I_{\text{bat}}, \min \leq I_{\text{bat}} \leq I_{\text{bat}}, \max, \\
I_{\text{uc}}, \min \leq I_{\text{uc}} \leq I_{\text{uc}}, \\
4R_{\text{b}} \cdot P_{\text{uc}} \leq V_{\text{bat}}^2.
\end{array} \right.
\]

The optimization objective thus can be expressed as

\[
E_{\text{con}} = \min \sum (E_{\text{bat}}(t) + E_{\text{uc}}(t)),
\]

where

\[
E_{\text{bat}} = I_{\text{bat}}(t) \cdot [V_{\text{bat}}(t) \cdot \cos (\theta_{b}(t))] + R_{\text{bat}},
\]

Table 2: Continued.

|   | SSA | WOA | GA | MFO | PSO | DA |
|---|-----|-----|----|-----|-----|----|
| F19 | 7.47E-04 | 7.47E-04 | NA | 7.47E-04 | 7.47E-04 | 7.47E-04 |
| $p$ | 1 | 1 | NA | 1 | 1 | 1 |
| $zval$ | 3.37E+00 | 3.37E+00 | NA | 3.37E+00 | 3.37E+00 | 3.37E+00 |
| Rank sum | 145 | 145 | NA | 145 | 145 | 145 |
| F20 | 7.89E-01 | 1.33E-01 | NA | 7.65E-01 | 9.21E-02 | 9.38E-01 |
| $p$ | 0 | 0 | NA | 0 | 0 | 0 |
| $zval$ | 2.68E-01 | 1.50E+00 | NA | -2.99E-01 | 1.68E+00 | -7.72E-02 |
| Rank sum | 108.5 | 124 | NA | 101 | 126 | 103.5 |
| F21 | 0.6701 | 0.1675 | NA | 0.0435 | 0.3222 | 0.3662 |
| $p$ | 0 | 0 | NA | 1 | 0 | 0 |
| $zval$ | 0.426 | 1.3803 | NA | -2.0186 | -0.9899 | -0.9036 |
| Rank sum | 109 | 115 | NA | 81 | 94 | 95 |
| F22 | 1 | 0.5828 | NA | 0.5842 | 1 | 0.6264 |
| $p$ | 0 | 0 | NA | 0 | 0 | 0 |
| $zval$ | 0 | -0.5493 | NA | -0.5472 | 0 | -0.4869 |
| Rank sum | 105.5 | 100 | NA | 100 | 105 | 100.5 |
| F23 | NA | NA | NA | 0.0348 | 0.3681 | 1.59E-05 |
| $p$ | 0 | 1 | NA | 1 | 0 | 1 |
| $zval$ | NA | NA | NA | -2.1102 | -0.9 | 4.3153 |
| Rank sum | 105 | 106 | NA | 85 | 100 | 155 |
\[ \text{Euc} = I_{\text{uc}}(t) \ast [(V_{\text{uc}}(t) \ast \cos(\theta_{\text{uc}}(t))) + R_{\text{uc}}]. \]  

\( \theta_b \) and \( \theta_{\text{uc}} \) are the current phase angle of the currents \( I_{\text{bat}} \) and \( I_{\text{uc}} \), respectively.

4. Data Collection

4.1. Benchmark Functions. To analyze the performance of the MSSSA, it is essential to collect its performance data and that of standard metaheuristic optimization algorithms such as PSO, DA, GA, and WOA.

A comprehensive evaluation of the MSSSA on continuous domain problems is given by testing on a set of 23 benchmark functions. These functions are classified into three:

(i) Unimodal functions (F1-F7), which are high-dimensional functions with only one global minimum, thus relatively easier to solve compared to the rest of the test functions

(ii) High-dimensional multimodal functions (F8-F13) which have a number of local minima, thus the most difficult to solve

(iii) Low-dimensional multimodal functions (F14-F23) which have a smaller number of dimensions, thus fewer local minima compared to high-dimensional test functions [24]

The best scores of the selected algorithms were collected for each of the twenty-three standard benchmark functions over ten independent runs. The average score and standard deviation are then calculated to provide a comparison metric for the algorithm’s performance.

For fair comparison of the selected algorithms and the inclusion of high-dimensional multimodal functions, a high number of search agents and iterations are required [6] and for these tests which are given as

(i) No. of search agents = 60

(ii) Maximum iterations = 1000

Since they are stochastic algorithms, results obtained may significantly vary on every run. Performance can only be comparatively analyzed over a significant number of runs [6, 10, 11], for this study.

(iii) Number of = 1000

4.2. Dynamic Programming for HESS in EVs. Optimization of EV HESS control is a multistage decision-making problem, and these problems are split into subproblems solved consecutively in discrete time. To obtain the best control performance and compare the performances more accurately, dynamic programming is used for benchmark evaluation [25].

Theoretical minimum energy consumption over a period is calculated by simulating the energy demand for the HESS drive cycle. In this paper, to analyze the control strategy performance, the energy demand for a period of 120 seconds is generated from the speed and acceleration matrix as in Figure 6.
The EV energy demand is calculated by equation (16) resulting in the plot Figure 7 below. Optimization is done at every second instance as demand changes with time thus necessitating high speed performance for real time optimization. To minimize time taken for optimization, the max number iteration is set to 100, number of search agents $N = 60$, with the EV and HESS parameters given as follows:

- Vehicle mass $(m) = 3600$ (kg)
- Drag coefficient $(C_d) = 0.208$
- Air density $(Dens) = 1.225$ (kg/m$^3$)
- Reference area $(Area) = 2.34$ (m$^2$)
- UC capacitance $(Cuc) = 400$ (F)
- Battery voltage $(Vbat) = 350$ (V)
- Capacitor voltage $(Vuc) = 36$ (V)
- Battery internal resistance $(Rb) = 20000$ (Ω)
- Ultracapacitor internal resistance $(Ruc) = 700$ (Ω)
- Initial energy stored in battery $(Qbat) = 19800000000$ (J)
- Initial energy stored in ultracapacitor $(Quc) = 3600000$ (J)

The simulation is set for a minimal time duration since the SOC constraints have not been considered. To enforce current limitations, after initial optimization iteration, the battery current required is compared to the threshold maximum current from which the boundaries are adjusted as follows:

This is to ensure that energy from capacitors is only used to meet high power demand.

Data on energy consumption and current distribution were collected for 10 runs of the simulated drive cycle to analyze the theoretical performance of the MSSSA for HESS control.

5. Results, Analysis, and Discussion

In this section, the data collected based on the criterion specified in Section 4 are given and analyzed based on performance metrics for optimization algorithms and HESS control. These tests were performed on a HP EliteBook Folio 1020 G1 with the following:

- (i) Intel(R) Core (TM) M-5Y71 CPU @ 1.20GHz (4 CPUs)
- (ii) ~1.4 GHz Processor
- (iii) 8192 MB RAM

Results from the Benchmark tests were as summarised in Table 1 below. Comparing the average scores for the test functions, the MSSSA outperforms the SSA in functions F2, F3, F5, F7, F8, F10, F11, F12, and F13. Both are seen to achieve the global minimum for test functions F16, F22, and F23.

The Wilcoxon rank-sum test (a nonparametric test of the null hypothesis), i.e., for randomly generated sets of score values by two optimization algorithms $X$ and $Y$, the probability of $X > Y$ equals the probability $Y > X$. $p$ gives the $p$ values of a Wilcoxon rank sum test, and $h$ gives the logical value that indicates if the null hypothesis holds (0) or is rejected (1) at 5% significance level [26]. The results of the Wilcoxon rank sum test for the test function data of the MSSSA are given in Table 2 below.

The rank-sum test shows the MSSSA has lower median values for functions F2, F3, F7, F11, and F15 compared to the SSA. From the table, there is a small improvement of the MSSSA over the SSA in the rest of the benchmark functions scores. This is given by the rank sum value where a rank sum value less than 105 the MSSSA score is likely to be less than that of the algorithm in comparison; however, $h = 0$ as this is of less than 5% significance. For the low-dimensional multimodal functions F16, F20, F21, F22, and F23, the rank sum test shows no significant performance
Figure 9: Continued.
analytic data as majority of the select optimization algorithms result in global maxima for these functions.

Comparison of the mean scores of the MSSSA and the standard algorithms is given in a performance rank table, Table 3 given as follows.

5.1. EV HESS Control Simulation Results. To analyze the efficiency of an MSSSA-based control strategy for HESS in electric vehicles, the theoretical energy consumed and current distributions curves were obtained from the simulation as highlighted in Table 4.

Energy consumption is best optimized by the MSSSA as evident in the average values above. A visual representation of energy consumption over time is given by Figure 8.

An essential function of HESS control strategy apart from energy consumption optimization is the current distribution. A good control strategy maintains the battery discharge rate within the desired limits, reducing current transients and at the same time while keeping the capacitor discharge low at low power demand.

Figure 9 Below displays the current distribution curve over time for the metaheuristic optimization-based control strategies.

For MSSSA optimized control, battery current has less transients compared to the SSA, MFO, and DA. Better optimization of energy consumption means lesser demand at every state; thus, current limit constraint are exceeded fewer times. Overall, the MSSSA achieves the best energy distribution with the best optimization consistently outperforming the SSA, DA, PSO, and MFO algorithms.

Speed of the optimization process is critical for the viability of using metaheuristic algorithms as a control strategy. Ideally, this should take a fraction of a second with hardware limitations considered. Run time for the HESS control for each instance was recorded and a summary given in Table 5.

Table 5: HESS optimization speed test results.

|          | MSSSA   | SSA     | MFO     | PSO     | DA       |
|----------|---------|---------|---------|---------|----------|
| Min time (s) | 0.026658| 0.019028| 0.024243| 0.020805| 1.221294 |
| Max time (s) | 0.190726| 0.094531| 0.092077| 0.091801| 15.44288 |
| Avg.      | 0.055761| 0.043362| 0.056142| 0.051355| 5.450881 |
| Std. Dev. | 0.019776| 0.016242| 0.014043| 0.016732| 4.923087 |

Figure 9: (a) MSSSA current distribution. (b) SSA current distribution. (c) MFO current distribution. (d) PSO current distribution. (e) DA current distribution.
an optimization algorithm, the cost function is at the core of the MS-SSA function and design. From the benchmark test functions and EV HESS simulation, the MS-SSA returns the best and second-best score in a 14 out of 23 applicable test problems explored. This puts the functions performance compared to the other selected algorithms at 60%.

The algorithm introduces a dynamic learning rate, and the learning rate can be said to be random but fundamentally governed by the positional difference of the search agents relative to the global minima. This means that though random, that initially, the learning rate is large and immensely decreases as it approaches the minima.

Unlike conventional algorithms, the MS-SSA convergence rate varies with the number of iterations enabling it to minimize error improving performance particularly where limitation on number of iterations is needed. The coupling of this algorithm with the dynamic model for real-time optimization of EV HESS has the advantages of improved performance in terms of energy consumption over fuzzy control whilst being fast enough to meet the speed requirements.

6. Conclusion

This paper introduces a master-slave salp swarm algorithm for real-time control of hybrid energy storage Systems in pure electric vehicles. By introducing a master-slave learning scheme, the algorithms’ exploitation is improved, as the dynamic learning rate balances exploration and exploitation for improved global performance in a variety of optimization problems. From the results, we established that theoretically, MSSSA offers the best optimized control strategy for EV HESS where few iterations are necessary for real-time optimization and control.

This work provides an abstract model for incorporation of an improved metaheuristic algorithm in a real-time optimization problem. Practical implementation of the conceptualized model requires further research into different master-slave learning techniques suited to particular problems, predictive tools to further the potential for real-time optimization, and multiobjective algorithm design to encompass complex nature of real-time optimization problems.

Data Availability

The data used to derive the results of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no competing interests concerning the publication of this paper.

Acknowledgments

This work has been sanctioned and fully funded by the University of Nairobi, Department of Electrical and Information Engineering.

References

[1] I. Azizi and H. Radjeai, “A new strategy for battery and supercapacitor energy management for an urban electric vehicle," Electrical Engineering, vol. 100, no. 2, pp. 667–676, 2018.
[2] D. Bu and C. Zhu, "Research on the optimum continuous mileage under the pure electric mode of plug-in hybrid electric vehicles," Applied Mechanics and Materials, vol. 672-674, pp. 1179–1182, 2014.
[3] R. Bousmaha, R. M. Hamou, and A. Amine, "Optimizing Connection Weights in Neural Networks Using Hybrid Metaheuristics Algorithms," International Journal of Information Retrieval Research (IJIRR), vol. 12, no. 1, pp. 1–21, 2022.
[4] G. Amir, Y. Xin-She, T. Siamak, and A. Amir, "Metaheuristic algorithms in modelling and optimization," Metaheuristic applications in structures and infrastructures, vol. 1, no. 9780123983640, pp. 2–5, 2013.
[5] Z. Yaseen, H. Faris, and N. Al-Ansari, "Hybridized extreme learning machine model with salp swarm algorithm: a novel predictive model for hydrological application," Complexity, vol. 2020, Article ID 8206245, 14 pages, 2020.
[6] S. Mirjalili, A. Gandomi, S. Mirjalili, S. Saremi, H. Faris, and S. Mirjalili, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," Advances in Engineering Software, vol. 114, pp. 163–191, 2017.
[7] C. Magele and T. Ebner, Optimization in Electrical Engineering, Institute for Fundamentals and Theory in Electrical Engineering Technical University of Graz, Graz, 2021.
[8] Q. Duan, L. Wang, H. Kang, Y. Shen, X. Sun, and Q. Chen, "Improved Salp Swarm Algorithm with Simulated Annealing for Solving Engineering Optimization Problems," Symmetry, vol. 13, no. 6, p. 1092, 2021.
[9] B. Yang, Y. Chen, Z. Zhao, and Q. Han, "A master-slave particle swarm optimization algorithm for solving constrained optimization problems," in 2006 6th World Congress on Intelligent Control and Automation, vol. 1, pp. 3208–3212, Dalian, 2021.
[10] S. U. Mane, A. C. Adamuthe, and R. R. Omame, "Master-Slave TLBO Algorithm for Constrained Global Optimization Problems. EAI Endorsed Transactions on Scalable," Information Systems, vol. 8, no. 30, p. e2, 2020.
[11] E. Momanyi and D. Segera, "A master-slave binary grey wolf optimizer for optimal feature selection in biomedical data classification," BioMed Research International, vol. 2021, Article ID 5556941, 12 pages, 2021.
[12] K. Ye, P. Li, and H. Li, "Optimization of hybrid energy storage system control strategy for pure electric vehicle based on typical driving cycle," Mathematical Problems in Engineering, vol. 2020, Article ID 1365195, 12 pages, 2020.
[13] J. Li, Q. Zhou, Y. He et al., "Dual-loop online intelligent programming for driver-oriented predict energy management of plug-in hybrid electric vehicles," Applied Energy, vol. 253, article 113617, 2019.
[14] H. Zhou, P. Wei, and L. Sun, "Development status of electric vehicles," Journal of Asian Electric Vehicles, vol. 2, no. 1, pp. 531–534, 2004.
[15] J. Y. Qu, R. Wang, and Y. C. Zhou, "Study of EV Composite Power Supply for Energy Saving," in Advanced Materials Research, vol. 578, pp. 3–6, Trans Tech Publications Ltd., 2012.
[16] S. Lu, K. A. Corzine, and M. Ferdowsi, "A new battery/ultracapacitor energy storage system design and its motor drive
integration for hybrid electric vehicles,” *IEEE transactions on vehicular technology*, vol. 56, no. 4, pp. 1516–1523, 2007.

[17] H. Yu, R. Lu, T. Wang, and C. Zhu, “Battery/ultra-capacitor hybrid energy storage system used in hev,” *Journal of Asian Electric Vehicles*, vol. 8, no. 1, pp. 1351–1356, 2010.

[18] N. Schouten, M. Salman, and N. Kheir, “Energy management strategies for parallel hybrid vehicles using fuzzy logic,” *Control Engineering Practice*, vol. 11, no. 2, pp. 171–177, 2003.

[19] Q. Zhang, L. Wang, G. Li, and Y. Liu, “A real-time energy management control strategy for battery and supercapacitor hybrid energy storage systems of pure electric vehicles,” *Journal of Energy Storage*, vol. 31, article 101721, 2020.

[20] Y. Kawahara, “Understanding the underlying algorithms and theories of machine learning,” *Impact*, vol. 2020, no. 2, pp. 40–42, 2020.

[21] J. Sanguesa, V. Torres-Sanz, P. Garrido, F. Martinez, and J. Marquez-Barja, "A Review on electric vehicles: technologies and challenges," *Smart Cities*, vol. 4, no. 1, pp. 372–404, 2021.

[22] A. Burke, “Ultracapacitor technologies and application in hybrid and electric vehicles,” *International Journal of Energy Research*, vol. 34, no. 2, pp. 133–151, 2010.

[23] W. B. Blakemore, C. D. Nash, and W. Perryman, *Physical Forces Affecting an Automobile in Motion*, Texas Department of Public Safety, Austin, 1st ed. edition, 1961.

[24] K. Hussain, M. Mohd Salleh, S. Cheng, and R. Naseem, "Common benchmark functions for metaheuristic evaluation: a review," *JOIV: International Journal on Informatics Visualization*, vol. 1, no. 4-2, pp. 218–223, 2017.

[25] R. Wang and S. M. Lukic, “Dynamic programming technique in hybrid electric vehicle optimization,” in *2012 IEEE International Electric Vehicle Conference*, pp. 1–8, Greenville, SC, USA, 2012.

[26] E. West, “The Signed-Rank (Wilcoxon) Test,” *The Lancet*, vol. 293, no. 7593, pp. 526–529, 1969.

[27] Z. Liu and T. Nishi, “Multipopulation ensemble particle swarm optimizer for engineering design problems,” *Mathematical Problems in Engineering*, vol. 2020, Article ID 1450985, 30 pages, 2020.