IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON EVOLVING TECHNOLOGIES IN ENERGY STORAGE SYSTEMS FOR ENERGY SYSTEMS APPLICATIONS

Received December 23, 2020, accepted December 27, 2020, date of publication January 13, 2021, date of current version January 26, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051452

Optimal Battery Energy Storage System Scheduling Based on Mutation-Improved Grey Wolf Optimizer Using GPU-Accelerated Load Flow in Active Distribution Networks

DORIAN O. SIDEA, (Member, IEEE), IRINA I. PICIOROAGA, (Student Member, IEEE), AND CONSTANTIN BULAC, (Member, IEEE)

Department of Electrical Power Systems, University "Politehnica" of Bucharest, Bucharest 060042, Romania

Corresponding author: Dorian O. Sidea (dorian.sidea@upb.ro)

This work was supported by the University "Politehnica" of Bucharest.

ABSTRACT In this paper, a novel Mutation-Improved Grey Wolf Optimizer (MIGWO) model is introduced in order to solve the optimal scheduling problem for battery energy storage systems (BESS), considering the mass integration of renewable energy sources (RES), such as solar and wind generation, in active distribution networks. In this regard, four improvements are applied to the conventional GWO algorithm to modify the exploration–exploitation balance for an enhanced convergence rate. The validity and performance of the proposed model are tested on 23 classical benchmark functions and compared to the original algorithm. The new technologies present in active distribution networks lead to increased complexity in the efficient coordination of existing resources, making it necessary to resort to advanced optimization and calculation methods. As operational planning and control functions in power systems are computationally demanding and require multiple power flow calculations, the necessity of simultaneous (parallel) computing techniques emerged. In order to reduce the computing time, an accelerated GPU parallel computing technique is also applied in the proposed model. The MIGWO algorithm is further applied on the modified IEEE-33 bus system aiming to minimize the total power losses, based on the optimal coordination of BESS operation scheduling and RES generation for multiple load demand and local generation scenarios, as well as for various initial state-of-charge values of BESS.

INDEX TERMS Active distribution network, battery energy storage system, grey wolf optimization, parallel computing.

NOMENCLATURE

ABBREVIATIONS

ADN	Active distribution network
BESS	Battery energy storage system
CPU	Central processing unit
GPU	Graphical processing unit
PV	Photovoltaic panels
RES	Renewable energy sources
SOC	State-of-charge
WT	Wind turbine

SETS

t	Index of time
i, j, k	Index of buses
ij	Index of lines
l	Index of MIGWO/GWO iterations
T	Number of time intervals
N_B	Set of buses
N_L	Set of lines
N_{mut}	Number of mutant wolves
N_{tot}	Total number of wolves
N_β	Number of β wolves
N_δ	Number of δ wolves
N_ω	Number of ω wolves

PARAMETERS

W_{Bmax}	BESS energy capacity
SOC_0	Initial state-of-charge
SOC_{min}, SOC_{max}	Lower and upper bounds of the BESS state-of-charge

The associate editor coordinating the review of this manuscript and approving it for publication was Eklas Hossain.

ε_{SOC}	Daily SOC balance acceptable error
$P_{B,min}$, $P_{B,max}$	Lower and upper bounds of the BESS power exchange
η_{ch}, η_{dsc}	BESS charging and discharging efficiencies
Δt	Length of time interval
V_S	Source bus operational voltage
P_{Gi}, Q_{Gi}	Active and reactive power generated at bus i
$\underline{S}_{Li}$, P_{Li}, Q_{Li}	Apparent, active and reactive power demand at bus i
$\underline{Z}_{ij}$, R_{ij}, X_{ij}	Impedance, resistance and reactance of line ij
G_{ik}, B_{ik}	The real and imaginary part of the bus admittance matrix ik^{th} term
$V_{min,k}$, $V_{max,k}$	Lower and upper bounds of the voltage at bus k
$I_{max,ij}$	The maximum capacity of line ij
N_{mut}^{min}, N_{mut}^{max}	The minimum and maximum number of mutants

VARIABLES

$P_{B,t}$	BESS active power exchange during the t^{th} time interval
SOC_t	The state of charge during the t^{th} time interval
ΔSOC_t	The state of charge variation during the t^{th} time interval
$\Delta P_{tot,t}$	The total active power losses
$P_{s,t}$	The total active power supplied by the slack bus
θ_i	The bus voltage angle at bus i
V_i	The bus voltage magnitude at bus i
$I_{ij,t}$	The current magnitude for line ij during the t^{th} time interval

I. INTRODUCTION

The modern-day electrical distribution systems are embracing the use of renewable energy sources for their environmental and economic benefits. The climate concerns that have arisen in the past decades encouraged the massive renewable energy sources integration, such as wind turbines (WT), photovoltaics (PV) and biomass, as clean power supply solutions. Hence, current policies around the world promote the renewable sources installation to respond to increasing demand challenges and environmental issues. On that account, a new framework for the energy distribution infrastructure took form, namely the active distribution networks (ADN), that incorporates advanced monitoring techniques and management strategies [1]. Based on the dropping generation cost, technologies improvement and governmental incentives, WTs and PVs are particularly expected to cover a significant part of the global energy portfolio in the following years. However, the main drawback of these resources, consisting in permanent fluctuations of the output, led to the re-emergence of the energy storage concepts [2], [3]. Among these, the battery energy storage systems (BESS) have known a high interest and development in the last decades, due to their stability, accessibility and ease of control, which encouraged their deployment in most the power system sectors. The BESS capability of shifting in time the power load and generation stimulated their implementation in providing a wide range of services to the power grid [4]. In [5]-[7], the benefits of energy storage in managing decentralized renewable energy resources and meeting the load demand reliability are analyzed in the ADN context, based on the BESS optimal operation. The BESS capacities in frequency regulation were also studied. Authors of [8] consider the increasing share of energy storage in frequency regulation markets in order to develop an optimized bidding policy, that maximizes profits while satisfying the market requirements. In [9], control strategies for frequency regulation are proposed by parallel operation of multiple energy storage systems, based on their state of charge and ramping capabilities, while [10] emphasizes the importance of BESS in frequency stability for low inertia power systems. Climatic changes have also amplified concerns regarding power systems security. The intensified severity and frequency of natural disasters, including hurricanes, floods, ice storms, wildfires etc., have become the main cause of multiple damages of power systems and extended blackouts. Thus, a new concept of reliability was introduced, namely resilience. Concerning particularly phenomena with low probability, but severe socially and economically impacts, more and more studies are currently focusing on developing strategies for resilience enhancement in the energy infrastructure. A proactive operation strategy for hybrid microgrids is proposed in [11], that assures the survivability of critical loads using RES and BESS units. The impact of cost-optimal coordination of PV and storage systems is analyzed for commercial buildings resilience in [12], while [13] investigates a variety of battery systems in order to identify the optimal technology for economic and resilience perspectives.

The new technologies integration leads to increased complexity in the efficient coordination of existing resources in active distribution networks, urging the resort to advanced optimization methods. Based on their capability of solving large-scale linear and nonlinear problems, meta-heuristic optimization techniques grew in popularity in the past decades. Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and others are well-known both in computer science applications and power systems operation. Introduced in [14] in 2014, the GWO algorithm represents a novel swarm-based meta-heuristic technique inspired by the hunting behavior of the grey wolfs. Given its great balance between exploration and exploitation, this technique presents better convergence characteristics compared to other optimization methods. Based on these features, GWO has been frequently approached in the past years for solving various optimization problems in different disciplines. Applications of GWO can also be found in dealing with nonlinear engineering problems specific to

power systems, from optimal power flow [15], economic load dispatch [16] to the optimal Volt/Var control [17]. In order to get the best features of the algorithm, modified versions have been analyzed in studies, such as [18], where the exploration–exploitation balance of the standard GWO is modified in order to obtain an enhanced convergence rate. Hybrid methods have also been approached. In [19], a Fuzzy Logic-GWO method is implemented for a multi-objective intelligent energy management in grid-connected microgrids. Authors of [20] introduce a hybrid PSO-GWO algorithm that provides better results compared to the two standard methods and apply this framework to solve the optimal power flow problem.

The penetration of the unregulated RES and the decentralized operation tendencies in current power networks requires more and more computational capacity for more complex system analysis. In order to reduce the computation time for large-scale problems, several studies have been conducted on the serial execution of optimization problems calculation in power systems. However, most of these problems require multiple power flow computations. This time-consuming process can be shortened using simultaneous (parallel) computing techniques. Considering the hundreds of simultaneous power flow calculations that should be performed, the GPU-CPU heterogeneous architectures can prove themselves an efficient alternative in parallel programming [21]. Ever since GPUs emergence, several studies have focused on migrating parallel computing from CPU (Central Processing Unit) to GPU (Graphical Processing Unit), which demonstrates a great potential in applications specific to various research disciplines and industrial fields [22], including the energy sector.

The study presented in this paper focuses on determining the optimal short-term operational scheduling of a BESS, based on the local available RES and the energy demand in an active distribution network, that aims at total power losses minimization. In this regard, the authors propose a Mutation-Improved GWO algorithm to overcome the risk of convergence in local optimums of the standard technique. The study exploits the benefits of GPU accelerated global search, in order to explore the large and complex search space of the optimal BESS scheduling problem for better solutions. The proposed algorithm is applied on the IEEE 33-bus distribution system modified to include multiple renewable energy sources (WTs and PVs) and a storage device.

The paper structure is organized as follows. Section II illustrates the problem formulation for the optimal BESS scheduling for power losses minimization in ADN. Section III is devoted to the optimization model presentation. Here, the standard GWO algorithm and the new improvements proposed by the authors are briefly presented, as well as the load flow calculation and the model implementation. The analyzed network and data are described in Section IV, as well as the numerical results and discussions. Finally, Section V concludes the paper.

II. PROBLEM FORMULATION

As system operators are generally responsible for ensuring the efficiency and reliability of network operation, minimizing active losses is considered when adopting certain operating strategies. As RES prove themselves more economically and viable sources, the battery energy storage systems become an indispensable equipment in the proper functioning of the electricity systems. Assuming an accurate forecast of the demand and renewable resources production, the scope of the optimal BESS operation scheduling is to minimize the active power losses, while satisfying the network technical constraints.

The generic power losses minimization problem can be formulated as follows:

$$\begin{aligned} &\text{Minimize } f\,(x,\mu)\\ &\text{subject to: } g\,(x,\mu)=0\\ &\qquad\qquad h\,(x,\mu)\le 0 \end{aligned} \tag{1}$$

where: f is the objective function to be minimized, g is the set of equality constraints, h are the inequality constraints, x is the control variables vector and μ is the system parameters vector. Given the purpose of the study to determine the optimal coordination of BESS and RES operation, the control variables vector x consists in the active power exchange of the storage system, as explicitly presented in the battery modeling section. The parameter vector μ includes: 1) the BESS parameters: energy capacity (W_{Bmax}), charging and discharging efficiencies (η_{ch} and η_{dsc}) and initial state-of-charge (SOC_0); 2) distribution network parameters: electrical lines resistance and reactance (R_{ij} and X_{ij}) and topology; and 3) operational data: source bus operational voltage V_S, active and reactive power demanded by the loads (P_{Li} and Q_{Li}) and generated by the distributed sources (P_{Gi} and Q_{Gi}) at each bus i during each time interval t.

A. BATTERY STORAGE MODELING

The optimization problem aims at performing the optimal scheduling of a battery storage system during a total of T time intervals. Consequently, the control variables vector x consists in the BESS active power exchange $P_{B,t}$ for each time interval.

$$x=\left[P_{B,1},P_{B,2}\ldots P_{B,t}\ldots P_{B,T}\right] \tag{2}$$

In the battery model, the passive sign convention is considered for the active power exchange between the BESS and the distribution network, so that for $P_{B,t}\ge 0$ the battery is charging, while for $P_{B,t}<0$ the battery is discharging. The state of charge at the end of the t^{th} time interval (SOC_t) is obtained based on the SOC at the end of the previous interval (SOC_{t-1}) to which the state of charge variation during the t^{th} time interval (ΔSOC_t) is added, as defined in the following equation:

$$SOC_t=SOC_{t-1}+\Delta SOC_t \tag{3}$$

The charging and discharging efficiencies η_{ch} and η_{dsc} are also considered within the BESS model, thus the state

of charge variation during the t^{th} time interval (ΔSOC_t) is expressed as follows:

$$\Delta SOC_t = \frac{\Delta W_t}{W_{Bmax}} = \begin{cases} \dfrac{P_{B,t} \cdot \eta_{ch} \cdot \Delta t}{W_{Bmax}} & \text{if } P_{B,t} \geq 0 \\ \dfrac{P_{B,t} \cdot \Delta t}{W_{Bmax} \cdot \eta_{dsc}} & \text{if } P_{B,t} < 0 \end{cases} \quad (4)$$

B. OBJECTIVE FUNCTION

The objective function f represents the total active power losses during a time period consisting in a number of T time intervals and is expressed as follows:

$$f(x) = \sum_{t=1}^{T} \Delta P_{tot,t} \quad (5)$$

The total active power losses, $\Delta P_{tot,t}$, for a given time interval t, are determined based on the active power balance as sum of the total active power supplied by the slack bus ($P_{s,t}$) and the net active powers at each bus (computed as the difference between distributed generation, $P_{Gi,t}$, and the load, $P_{Li,t}$), from which BESS active power exchange, $P_{B,t}$, is subtracted.

$$\Delta P_{tot,t} = P_{S,t} + \sum_{i \in N_B} \left(P_{Gi,t} - P_{Li,t}\right) - P_{B,t} \quad (6)$$

C. EQUALITY CONSTRAINTS

The equality constraints g from (1) consist of the typical active and reactive power flow equations, applied for each bus i as follows:

$$\begin{aligned} P_{Gi} - P_{Li} - V_i \sum_{k \in N_B} V_k \left[G_{ik} \cos(\theta_i - \theta_k) + B_{ik} \sin(\theta_i - \theta_k)\right] = 0 \\ Q_{Gi} - Q_{Li} - V_i \sum_{k \in N_B} V_k \left[G_{ik} \sin(\theta_i - \theta_k) - B_{ik} \cos(\theta_i - \theta_k)\right] = 0 \end{aligned} \quad (7)$$

where N_B is the set of buses in the distribution system, V_i and V_k are the bus voltage magnitudes at buses i and k, θ_i and θ_k are the bus voltage angles at buses i and k, G_{ik} and B_{ik} are the real and imaginary part of the bus admittance matrix ik^{th} term.

D. INEQUALITY CONSTRAINTS

The set of inequality constraints h from (1) includes the operation constraints of the storage system, but also the operational boundaries of the distribution network (the nodal voltage limits and the transmission capacity of the lines).

1) BESS OPERATION CONSTRAINTS

Three types of operation limits are considered for the BESS: the charging and discharging power boundaries, the energy capacity limits and the charging-discharging balance. The active power exchange between the battery and the distribution system, $P_{B,t}$, has to be maintained within the imposed lower and upper bounds ($P_{B,min}$ and $P_{B,max}$) for each time interval t, as denoted by:

$$P_{B,min} \leq P_{B,t} \leq P_{B,max} \quad (8)$$

Also, the BESS state of charge SOC_t must be kept within its minimum and maximum limits, SOC_{min} and SOC_{max}, that are imposed by:

$$SOC_{min} \leq SOC_t \leq SOC_{max} \quad (9)$$

For the optimization framework presented in this paper, the charging/discharging balance is expressed as the absolute value of the difference between the initial state of charge, SOC_0, and the final state of charge, SOC_T, which must be maintained at a lower value than the SOC balance acceptable error, ε_{SOC}:

$$|SOC_0 - SOC_T| \leq \varepsilon_{SOC} \quad (10)$$

2) DISTRIBUTION NETWORK OPERATION CONSTRAINTS

As the objective function aims at improving the distribution network operation during T time intervals, the load flow calculation is performed for each t time interval. Consequently, the distribution network operation constraints are imposed in order to assure, during each time interval t, that all bus voltages $V_{k,t}$ are within the acceptable range $[V_{min,k}; V_{max,k}]$, while all line currents, $I_{ij,t}$, do not exceed the capacity limits ($I_{max,ij}$).

$$V_{min,k} \leq V_{k,t} \leq V_{max,k}\ k \in N_B, \quad t = 1, 2 \ldots T \quad (11)$$

$$I_{ij,t} \leq I_{max,ij}\ ij \in N_L, \quad t = 1, 2 \ldots T \quad (12)$$

E. INTEGRATION OF EQUALITY AND INEQUALITY CONSTRAINTS

As this study employs a meta-heuristic solver, several specific implementation methods are necessary to be applied in order to integrate the equality and inequality constraints comprised within the optimization model.

In order to ensure the compliance of the equality constraints g given in (7), the load flow calculation is performed within the objective function, by using the Backward-Forward Sweep Method.

The inequality constraint (8) regarding the BESS active power limits, $P_{B,t}$, is modeled within the meta-heuristic algorithm by using the $P_{B,min}$ and $P_{B,max}$ limits as boundaries for the control variables vector x.

The other inequality constraints, (9) – (12), are integrated as penalty functions. In this manner, the implemented objective function, F, is defined by adding four penalty functions (P_1, P_2, P_3 and P_4), with corresponding weights (w_1, w_2, w_3 and w_4), to the initial objective function f.

$$F(x) = w_0 \cdot f + w_1 \cdot P_1 + w_2 \cdot P_2 + w_3 \cdot P_3 + w_4 \cdot P_4 \quad (13)$$

As meta-heuristic algorithms rely on randomly generated individuals, the possibility to encounter divergent load flow scenarios must be taken into consideration. In this regard, an additional penalty coefficient w_0 is applied to the objective

function f. In this manner, the individuals described by divergent load flows are penalized using a high w_0 value, while for the individuals corresponding to convergent load flows $w_0 = 1$.

In order to ensure the compliance with the two inequality constraints (9) and (10) regarding the battery state of charge limits, the first two penalty functions, P_1 and P_2, are introduced. The penalty function P_1 bounds the SOC values within the minimum and maximum limits during the day and is determined as sum of the penalty coefficients $p_{1,t}$, defined for every t time interval during the analyzed period:

$$P_1 = \sum_{t=1}^{T} p_{1,t} \tag{14}$$

If a SOC limit is violated, the penalty coefficient $p_{1,t}$ is equal to the positive difference between the SOC value and the exceeded limit. Otherwise, if the SOC value is between limits, the penalty coefficient equals zero.

$$p_{1,t} = \begin{cases} SOC_{min} - SOC_t, & \text{if } SOC_t < SOC_{min} \\ 0, & \text{if } SOC_{min} \le SOC_t \le SOC_{max} \\ SOC_t - SOC_{max}, & \text{if } SOC_t > SOC_{max} \end{cases} \tag{15}$$

The second penalty function, P_2, is defined in order to enforce the constraint regarding the BESS charging-discharging balance denoted by (10). In this purpose, P_2 equals the absolute value of the difference between the final and initial SOC values if this value is greater than the acceptable SOC error ε_{SOC}, and zero otherwise.

$$P_2 = \begin{cases} |SOC_0 - SOC_T|, & \text{if } |SOC_0 - SOC_T| > \varepsilon_{SOC} \\ 0, & \text{if } |SOC_0 - SOC_T| \le \varepsilon_{SOC} \end{cases} \tag{16}$$

The two inequality constraints regarding the distribution network operation are implemented by using the penalty functions P_3 and P_4. The P_3 penalty function enforces the voltage limits constraint (11), and is defined as follows:

$$P_3 = \sum_{t=1}^{T} \sum_{k \in N_B} p_{3,k,t} \tag{17}$$

where $p_{3,k,t}$ represents the penalty coefficient for the bus k, during the t time interval.

$$p_{3,k,t} = \begin{cases} V_{min} - V_{k,t}, & \text{if } V_{k,t} < V_{min} \\ 0, & \text{if } V_{min} \le V_{k,t} \le V_{max} \\ V_{k,t} - V_{max}, & \text{if } V_{k,t} > V_{max} \end{cases} \tag{18}$$

The line capacity constraint (12) is implemented by using the P_4 penalty function:

$$P_4 = \sum_{t=1}^{T} \sum_{ij \in N_L} p_{4,ij,t} \tag{19}$$

where N_L is the set of electrical lines and $p_{4,ij,t}$ the penalty coefficient for the line ij, during the t time interval:

$$p_{4,ij,t} = \begin{cases} I_{ij,t} - I_{\max,ij}, & \text{if } I_{ij,t} > I_{\max,ij} \\ 0, & \text{if } I_{ij,t} \le I_{\max,ij} \end{cases} \tag{20}$$

III. OPTIMIZATION MODEL STRUCTURE

A. THE STANDARD GWO OVERVIEW

The grey wolf optimizer algorithm has been developed by Mirjalili *et al.* in [14], based on the social behavior and hunting techniques of the grey wolf belonging to Canidae family. The pack hierarchy consists of a leader named alpha (α) and three subordinated layers composed of: beta wolves (β) that assist the leader in decision making, and delta wolves (δ) that ensure the leading of the last layer in the hierarchy, namely the omega wolves (ω). In order to mathematically formulate the social behavior that describes the GWO algorithm, from the four hierarchical groups, the α, β and δ denote the first three best solutions over the course of iterations, while the rest of the possible solutions are defined as ω. The simplicity of the concept makes it easy to implement for complex computationally problems, which justifies the preferences regarding the GWO use over other current metaheuristic methods [23].

1) SURROUNDING THE PREY

The mathematical formulation of the prey encircling process is presented below:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(l) - \vec{X}(l) \right| \tag{21}$$

$$\vec{X}(l+1) = \vec{X}_p(l) - \vec{A} \cdot \vec{D} \tag{22}$$

where l represents the current iteration, $\vec{X}_p(l)$ is the position vector of the prey, while $\vec{A}$ and $\vec{C}$ are coefficient vectors given by:

$$\vec{A} = 2 \cdot \vec{a} \cdot \vec{r}_1 - \vec{a} \tag{23}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{24}$$

The elements of vector $\vec{a}$ are linearly decreased from value 2 to 0 over the course of iterations, while the components of $\vec{r}_1$ and $\vec{r}_2$ are randomly selected between 0 and 1.

2) THE HUNTING

Depending on the first three best solutions obtained and saved so far (α, β and δ), the other search agents (including the omegas) update their positions in the search space according to the position of the best search agents, using the following equations:

$$\begin{aligned} \vec{D}_\alpha &= \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \rightarrow \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{D}_\beta &= \left| \vec{C}_2 \cdot \vec{X}_\beta - \vec{X} \right| \rightarrow \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{D}_\delta &= \left| \vec{C}_3 \cdot \vec{X}_\delta - \vec{X} \right| \rightarrow \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \end{aligned} \tag{25}$$

$$\vec{X}(l+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{26}$$

In the previous formulas, the GWO is obliged to explore and exploit the search space based on the components of vectors $\vec{A}$ and $\vec{C}$.

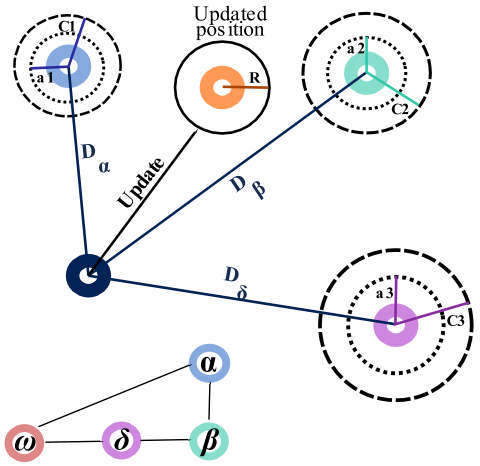

FIGURE 1. Position updating process of each wolf, considering the positions of the alfa, beta and delta wolves. Source: [14].

During exploration, the algorithm aims to discover new areas in the search space by applying changes on the current solutions, in order to find more promising ones (Fig.1). On the other hand, the exploitation process aims to improve the solutions identified during exploration, by investigating the neighborhood of each of these solutions. Based on $\vec{A}$, half of the iterative calculation is dedicated to exploration ($|\vec{A} \geq 1|$), while the other half is devoted to the exploitation process ($|\vec{A} < 1|$). Another controlled parameter is defined by vector $\vec{C}$, which contains random values in the range of [0; 2], that can adjust the contribution of the current position in defining the next position. For example, $\vec{C} > 1$ denotes a strong contribution, while for $\vec{C} < 1$, the effect in defining the next position is deemphasized. Such random behavior during the optimization process encourages exploration to avoid local optima.

B. PROPOSED IMPROVEMENTS

As authors of [14] suggested, improvements can be brought to the original GWO algorithm. This paper proposes four performance improvements for the standard GWO, that mainly aim to adjust the exploration process in order to reduce the probability for the algorithm stagnation in local minima.

1) SOC - FEASIBLE INITIAL POPULATION

The initial population influence upon the performance of population-based heuristic optimization algorithms – which also include GWO – is acknowledged by both recent [24]–[26] and older papers [27], [28]. A preprocessed initial population leads to faster convergence and better chances of finding performant solutions [29].

The main challenge regarding the initial population for the optimal BESS scheduling problem is represented by the state of charge constraints, (9) and (10), which should be implemented as penalty functions. The standard strategy, of generating an initial population of totally random individuals, generates a great fraction of initial wolves that violate these two restrictions. Numerous unfeasible individuals lead to a decrease in both convergence speed and performance. Therefore, two methodologies for generating a feasible initial population are proposed in this study.

The first strategy focuses on generating individuals that satisfy only the first restriction regarding the minimum and maximum SOC limits and consists in the procedure presented below. For every time interval t, the SOC value is calculated using (27) and the corresponding variable $P_{B,t}$ boundaries are determined, in order to enforce the SOC limits:

– The upper limit ($P_{B,t}^{max}$) is fixed as the minimum between the default upper boundaries ($P_{B,max}$) and the difference between SOC_{max} and SOC_t.
– The lower limit ($P_{B,t}^{min}$) is fixed as the maximum between the default lower boundaries ($P_{B,min}$) and the difference between SOC_{min} and SOC_t.

The algorithm also takes into account the BESS charging and discharging efficiencies, which divide or multiply the limits calculated above.

$$\begin{aligned} P_{B,t}^{min} &= \max\left\{W_{B,max}\left(SOC_{min} - SOC_t\right) \cdot \eta_{dsc}/\Delta t, P_{B,min}\right\} \\ P_{B,t}^{max} &= \min\left\{W_{B,max}\left(SOC_{max} - SOC_t\right)/\left(\eta_{ch} \cdot \Delta t\right), P_{B,max}\right\} \end{aligned} \tag{27}$$

The variable $P_{B,t}$ is further randomly generated between the imposed limits and the algorithm proceeds to the next time interval, $t + 1$.

$$P_{B,t}^{min} \leq P_{B,t} \leq P_{B,t}^{max} \tag{28}$$

The second strategy aims to generate individuals that satisfy both SOC restrictions. The main challenge in order to enforce the second restriction is the necessity to generate N random numbers whose sum (that also considers the charging and discharging efficiencies) is less than or equal to ε_{SOC}. The difficulty of the task consists of the possibility of generating random numbers that do not normally satisfy the second restriction. In this case, the necessary adjustments could either be easy to implement at the cost of reducing the diversity of the individuals, either difficult to implement while maintaining the random nature of the generation. Therefore, the accessible solution proposed by the authors in this paper consists in generating a larger number of individuals that comply with the first SOC constraint, further selecting only those who also satisfy the second constraint.

An empirical analysis, consisting in one million tests for three different total number of individuals (N_{tot}) of 10^2, 10^3 and 10^4, was conducted by the authors using parallel-accelerated computation. The number of feasible individuals obtained in each simulation (expressed in percentage relative to N_{tot}), are presented in Fig. 2, for

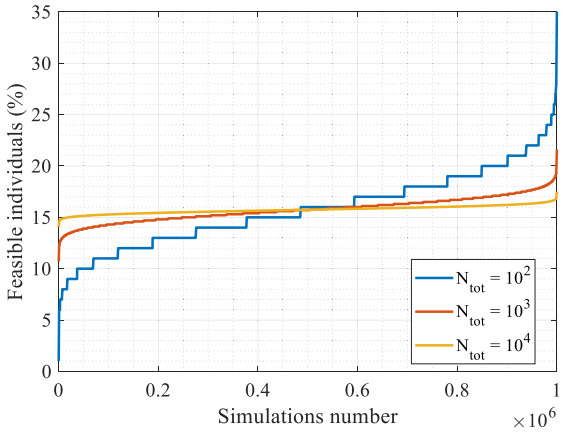

FIGURE 2. Number of feasible individuals obtained in one million simulations.

$\varepsilon_{SOC} = 5\%$. Based on these results, the authors concluded that, by applying the first strategy for a large number of individuals, fractions of 14.06% – 17.09% for $N_{tot} = 10^4$ and 10.7% – 21.7% for $N_{tot} = 10^3$ satisfy both SOC constraints. On the other hand, if this strategy is applied for a small number of individuals, it is necessary to generate at least $N_{tot} = 10^2$ individuals in order to obtain at least one feasible individual. For all three cases, the same average value of 15.73% was obtained for the percentage of feasible individuals.

2) MUTATION

Mirjalili *et al.* recommended in the paper that introduces the GWO algorithm the integration of evolutionary operators (such as crossover and mutation) as possible improvements of the standard algorithm performance [14]. This paper follows-up their suggestion and introduces the mutation operator. Specific to genetic algorithms, mutation ensures the population diversity by adding random alterations to the selected individuals. Therefore, a mutation operator, specifically adapted to the BESS scheduling problem, is added to the GWO algorithm in order to increase the exploration process performance.

During the first iterations, the GWO algorithm is focused on exploration, and as iterations proceed, the algorithm gradually switches the focus on exploitation. Since the mutation operator aims to boost the exploration, the number of mutant wolves that are generated is linearly decreasing over the iterative process to avoid interfering with the exploitation phase. The number of mutant wolves as a function of the current iteration is given in (29), where the minimum and maximum number of mutants (N_{mut}^{min} and N_{mut}^{max}) are expressed as percentage of the total number of wolves, N_{tot}.

$$N_{mut} = N_{tot} \cdot \left[\left(N_{mut}^{max} - N_{mut}^{min} \right) \cdot (1 - l_{act}/l_{\max}) + N_{mut}^{min} \right] \tag{29}$$

The mutation operator proposed in this paper randomly selects a parent wolf from the top 85% best individuals of the pack, operates a random alteration on it and returns it as a mutant wolf.

The parent wolf is selected with a probability of 15% as the alpha wolf, 15% as one of the beta or delta wolves or 70% from any of the rest top 85% wolves. Thereafter, one of the following alterations is randomly selected and applied on the original wolf:

- Randomly select one of the variables and replace it with a random generated value;
- Randomly select two variables and swap their values;
- Randomly select two variables and generate a small random number that is added to one variable and subtracted from the other;
- Generate a new random and feasible wolf;
- Add a small random quantity to a randomly selected variable;
- Add a small random number to a randomly selected variable and subtract the same random quantity divided by (n–1) from all other variables.

3) SURVIVAL OF THE FITTEST

The "survival of the fittest" (SOF) is one of the main principles that stand behind the development of all evolutionary algorithms and implies that better adapted individuals to the environment have better chances of survival. Inspired by [30], this paper proposes to integrate the opposite point of view of this principle as a mechanism to eliminate the least adapted individuals from the population. For this purpose, the wolves with the poorest performance (highest values of the objective function) are selected in order to be eliminated. The SOF mechanism is coordinated with the mutation operator, so that the mutant wolves will replace only the weakest individuals from the pack. Therefore, the number of eliminated individuals will be equal to the number of mutant wolves N_{mut} – meaning it will decrease linearly with the iterations.

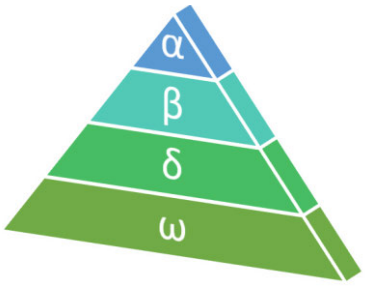

FIGURE 3. The hierarchical levels of the grey wolf pack (descendent dominance from top to bottom).

4) MULTIPLE β AND δ WOLVES

In the original GWO algorithm the hunting behavior of the pack is guided by the alpha, beta and delta wolves, which are selected as the top three ranking individuals within the pack (Fig. 3). Therefore, in some cases their positions may be very similar – or sometimes practically identical – which could

lead to a decreased performance of the exploration process. Consequently, this paper proposes the selection of multiple beta and delta wolves, in order to obtain the following social hierarchy: alpha wolf – the best individual; beta wolves – the following N_β top ranking wolves, and delta wolves – the following N_δ top ranking individuals. The updated wolf pack social hierarchy follows a pyramid-like structure: one alpha wolf, N_β number of beta wolves, N_δ delta wolves and omega wolves which constitute the rest of the pack; where $1 \leq N_\beta \leq N_\delta \leq N_\omega \leq N_{tot}$.

As iterations proceed, the number of beta and delta wolves decreases from the maximum values $(N_\beta^{max}, N_\delta^{max})$ to one, in order to avoid the exploitation process alteration during the final stages:

$$N_\beta = \max\left\{round\left\{N_\beta^{max} \cdot (1 - l_{act}/l_{\max})\right\}, 1\right\}$$
$$N_\delta = \max\left\{round\left\{N_\delta^{max} \cdot (1 - l_{act}/l_{\max})\right\}, 1\right\} \quad (30)$$

Although the proposed social hierarchy implies multiple beta and delta wolves, the mathematical model of the GWO algorithm is used in its original form with only the following addition: each beta and delta wolves, X_β and X_δ from (25), are randomly selected from the multiple beta and delta wolves.

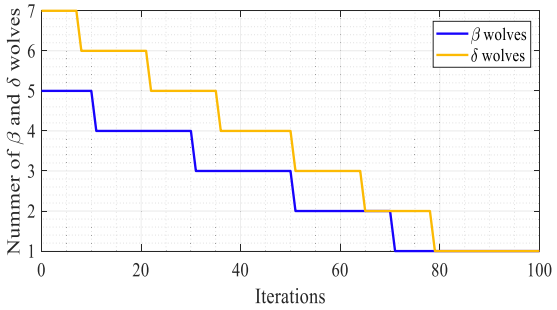

FIGURE 4. Number of beta and delta wolves variation with the iterations.

Fig. 4 shows the number of beta and delta wolves variation with the iterations, starting from $N_\beta = N_\beta^{max} = 5$ and $N_\delta = N_\delta^{max} = 7$, respectively, to $N_\beta = N_\delta = 1$, when reaching $l_{max} = 100$.

The improvements proposed by the authors and previously described are introduced within the original GWO algorithm, as illustrated in the flowchart of the resulting MIGWO algorithm depicted in Fig. 5. The steps corresponding to the four proposed improvements are marked by the blue border, while the green border indicates the accelerating methodology based on parallel computing used in this paper for the objective function calculation.

C. LOAD FLOW

Load flow (LF) calculation is a fundamental computation procedure performed on power systems in order to understand their operating conditions, by determining the steady-state parameters (i.e.: voltages, voltage angles, real power and reactive power). Several methods have been developed over the years for conducting power flow analysis [31].

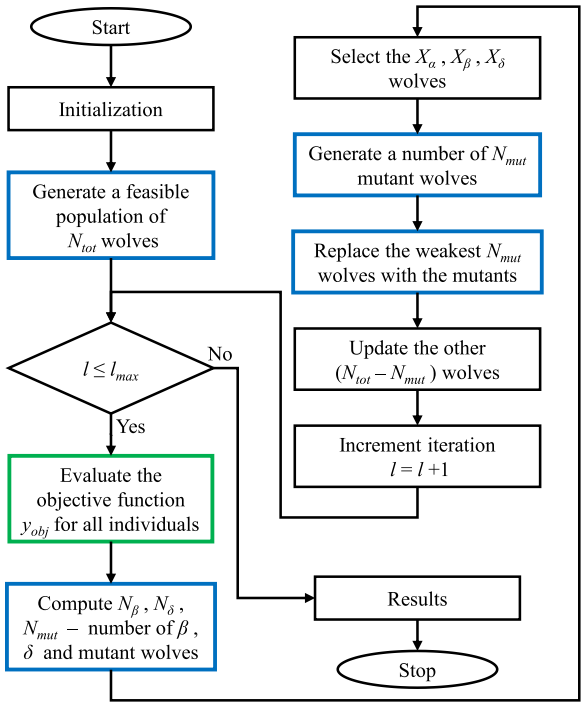

FIGURE 5. The MIGWO Flowchart.

Among them, the most popular include Newton-Raphson (NR), fast decoupled load flow (FDLF) and Seidel-Gauss (SG). However, the special features of distribution networks, such as high *R/X* ratios and radial/weakly meshed configurations, can lead to convergence problems for the previous mentioned methods. Therefore, the Backward-Forward Sweep Method (BFSM) had been developed for the distribution network LF analysis. Given the operation analysis of an active distribution network performed in this paper, BFSM is further used in the study.

For the BFSM implementation, a procedure that establishes the network buses analysis order is required. Based on the graph theory and considering the radial topology of the distribution network, a breadth-first search (BFS) algorithm is applied. BFS represents a uninformed search method that allows the systematically examination of all nodes in a graph, by exhaustively exploring the entire graph [32]. Starting with the root bus, the successor neighbours for each bus are identified, until all the buses are explored. Once all sets of neighbours (predecessors and successors) are known for each bus, the equations describing the BFSM are applied, as presented in the current section.

The BFSM is an iterative method, where each iteration consists of two computational steps:

a) *The backward sweep* – which implies the computing of current/power flows through the network branches according to the voltage updates. The calculation process starts from the end buses of the network and moves toward the source bus.
b) *The forward sweep* – which implies the voltage drop calculation. Bus voltages are updated in this step, starting

with the closest buses to the source (root bus) and forward sweeping to the end buses of the network. During this forward propagation, the currents/powers remain constant to the value obtained in the previous backward propagation.

The power flow calculation begins with the initialization step, where all bus voltages are considered equal to the source (root) bus voltage, V_S and the demand current is determined at each bus.

$$\underline{V}_j \simeq V_S, \quad \forall j \in N_B \tag{31}$$

$$\underline{I}_j \simeq \frac{\underline{S}^*_{L,j}}{\sqrt{3} \cdot V_S} \tag{32}$$

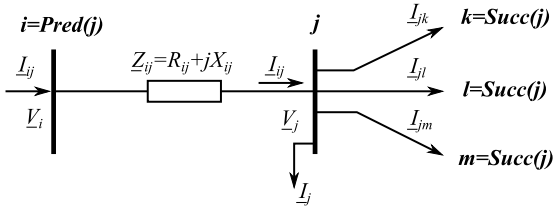

FIGURE 6. Node j and its connections to predecessor and successors.

Fig. 6 illustrates a node j from the distribution network, its predecessor node i, and its successor nodes grouped within the set $Succ(j)$. Note that each node (except the source) has a unique predecessor, while it may have one, none or multiple successors. Moreover, the current through the line that connects bus j with its predecessor i can be denoted $\underline{I}_{ij}$, as only one connection is possible.

During each iteration *iter* of the iterative process, the following steps are performed:

1) The backward sweep stage, the current demanded at each bus j ($\underline{I}_j$) is computed as the ratio between complex conjugates of the demanded apparent power ($\underline{S}_{L,j}$) and bus voltage ($\underline{V}_j$). Then the current through line ij ($\underline{I}_{ij}$) is determined as the sum between the demanded current at bus j ($\underline{I}_j$) and the sum of the currents through all its successor lines ($\underline{I}_{jk}$). The network is swept starting from the end busses and moving toward the source bus.

$$\underline{I}_j = \frac{\underline{S}^*_{L,j}}{\sqrt{3} \cdot \underline{V}^*_j} \tag{33}$$

$$\underline{I}_{ij} = \sum_{k \in Succ(j)} \underline{I}_{jk} + \underline{I}_j \tag{34}$$

2) During the forward sweep, the bus voltage for every bus j ($\underline{V}_j$) is computed as the difference between the bus voltage at its predecessor bus i ($\underline{V}_i$) and the voltage drop across the line that connects these busses ($\Delta\underline{V}_{ij}$). The latter is determined as the product between the impedance ($\underline{Z}_{ij}$) and the current ($\underline{I}_{ij}$) crossing the line that connects the i and j busses. In this stage, the network is swept from the source bus toward the ending busses.

$$\Delta\underline{V}_{ij} = \sqrt{3} \cdot \underline{Z}_{ij} \cdot \underline{I}_{ij} \tag{35}$$

$$\underline{V}_j = \underline{V}_i - \Delta\underline{V}_{ij} \tag{36}$$

3) The complex apparent power supplied by the source bus ($\underline{S}_S$) is determined as the product between the source bus voltage (V_S) and supplied current ($\underline{I}_S$) complex conjugate:

$$\underline{S}_S = \sqrt{3} \cdot V_S \cdot \underline{I}^*_S \tag{37}$$

4) Finally, a convergence test is performed by comparing the absolute value of the difference between the complex apparent power supplied by the source in the current and previous iterations. If the convergence test fails, then the iteration is increased ($iter = iter + 1$) and the iterative loop is repeated. Otherwise, the load flow calculation is completed, so the algorithm stops.

$$\left|\underline{S}^{(iter)}_S - \underline{S}^{(iter-1)}_S\right| \leq \varepsilon_{adm} \tag{38}$$

D. MODEL IMPLEMENTATION

As previously described, the optimization model aims to minimize the active power losses during an entire day by optimally scheduling the BESS operation on a per hour basis. Hence, the objective function evaluation requires the active power losses evaluation during each t time interval of a day, which implies that the load flow calculation must be performed for a number of T times for each individual within the wolf pack. The LF calculation for a distribution network is inherently time-consuming due to the necessity of using relatively complex iterative methods such as BFSM. Moreover, the necessity to perform a number of T load flow computations for each individual leads to a considerably long running time for the GWO algorithm. Consequently, this paper proposes to accelerate the LF calculation by using GPU parallel computing.

The optimization model is implemented by the authors in the MATLAB environment, which enables the users to take advantage of the GPU parallel-processing benefits, without demanding additional programming knowledge [33]. By using the Parallel Computing Toolbox™, the user can define a specific type of arrays, namely "gpuArrays", which are directly loaded into the GPU RAM and can be passed to over 500 built-in functions that will automatically run on the GPU.

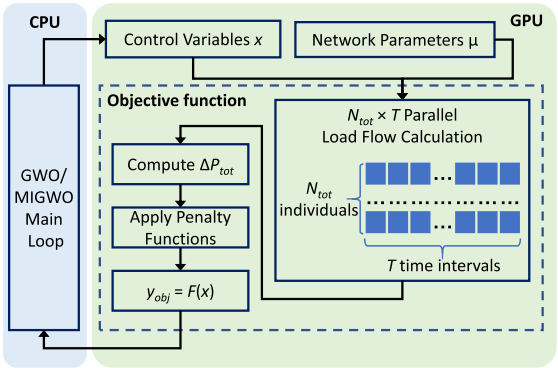

FIGURE 7. CPU and GPU computing flowchart.

A simplified flowchart is presented in Fig. 7 in order to highlight the optimization model CPU and GPU computing

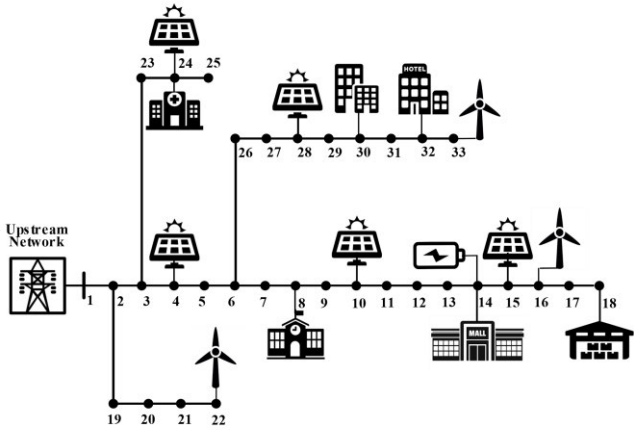

FIGURE 8. The modified IEEE 33-bus network.

structure. The GWO or MIGWO main loops are executed on the CPU, while GPU parallel computing is applied only for the objective function calculation.

The control variables vector, x, at the current GWO/MIGWO iteration is passed to the objective function f_{obj} and loaded into the GPU memory alongside the network parameters vector, μ. For the current iteration, the entire population requires a total number of $N_{tot} \times T$ load flow calculations, which are performed simultaneously by using GPU parallel-computed BFSM implementation. Then, the active power losses (ΔP_{tot}) are calculated and the four penalty functions are applied based on the resulted SOC, bus voltages and branch currents. Finally, the objective function values (y_{obj}) are gathered from the GPU memory and returned by the f_{obj} function to the GWO/MIGWO main loop.

IV. CASE STUDY

The applicability and efficiency of the proposed MIGWO model in solving the optimal coordination problem of renewable sources and storage systems was analyzed on a test distribution system. The ADN under study and the implemented data for RES generation and load demand are presented in Sections A and B. Section C of the case study focuses on the implemented load flow model validation and the performance comparison in terms of computation time between GPU and CPU. Also, the efficiency of the developed MIGWO model over the original GWO algorithm is tested in this section. Finally, the optimal day-ahead scheduling is conducted in Section D, while Section E is dedicated to a one-year analysis.

A. THE NETWORK UNDER STUDY

The algorithm proposed in this paper is applied on the modified IEEE 33-bus test network, whose original bus and branch data can be found in [34]. In order to model an active distribution network, multiple distributed resources are embedded in the grid, including wind turbines, photovoltaics and a storage device (Fig. 8).

B. LOAD AND DISTRIBUTED RESOURCES DATA

Most of the network buses are considered to supply residential loads, except for the following cases: a school (bus 8), a shopping mall (bus 14), a warehouse (bus 18), a hospital (bus 24), a medium office (bus 30) and a hotel (bus 32). The load profiles for the residential loads and for the various types of buildings considered in this study are specific to the geographical area of Atlanta, Georgia, USA, which is characterized by humid subtropical climate. The data for an entire year has been obtained from the OpenEI database [35] and processed so that the hourly profiles are expressed in percentage relative to the maximum active load, as depicted in Fig. 9.

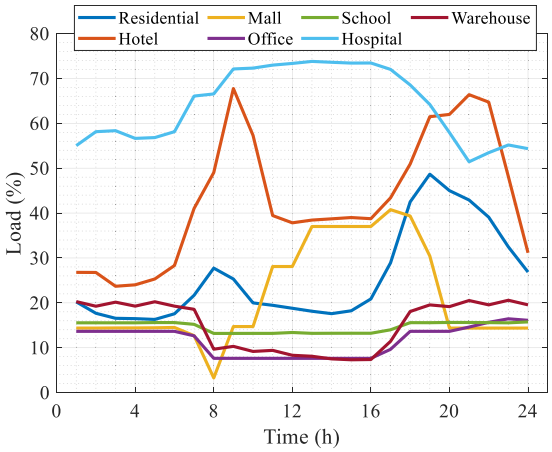

FIGURE 9. Example of daily load profiles (24-Dec-2019).

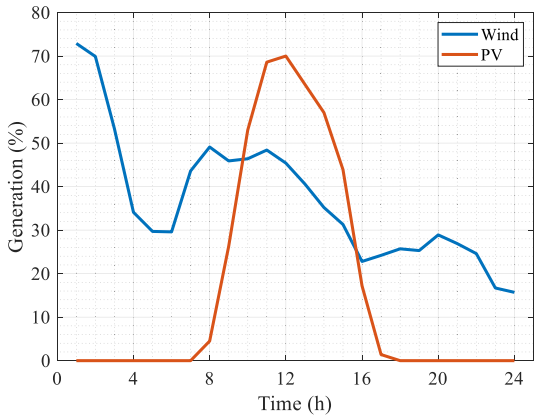

FIGURE 10. Example of wind and PV generation curves (24-Dec-2019).

Furthermore, a constant power factor is used to model the loads, therefore both active and reactive powers are obtained by applying the same percentage to the corresponding maximum values. For the simulations conducted in the case study, all the maximum active and reactive loads are increased by 1.6 times relative to the initial IEEE 33-bus network data, given the light load of the original grid.

The following distributed generation units have been added to the distribution network: three wind turbines each having

the rated active power of 1000 kW (connected at buses 16, 22 and 33, respectively) and five PV arrays with the rated active power of 200 kW each (connected to buses 4, 10, 15, 24 and 28, respectively). The hourly power output of the PVs and wind turbines for an entire year are obtained from Renewables Ninja database [36]–[38], considering the same geographical location as for the loads, namely Atlanta, GA. The original data are also processed in order to obtain an hourly profile expressed in percentage relative to the rated active power of each generation unit (Fig. 10). In this study, the DGs are considered to operate at constant inductive power factors of $\cos\varphi = 0.9$ for the wind turbines and $\cos\varphi = 0.95$ for the PVs. In this hypothesis, the active and reactive power of WTs and PVs are determined by applying the same percentage to the rated active and reactive power: 1000 kW and 484.3 kvar for the WTs and 200 kW and 65.7 kvar for the PVs.

The battery energy storage system, having the capacity (W_{Bmax}) of 1000 kWh, is installed at bus 14. Based on the economic benefits provided in terms of uninterrupted power supply even for small-to-medium scale applications, a lead-acid battery is considered as the storage technology applied in this study. The operational characteristics of the analyzed Pb-acid BESS extracted from [39] are presented in Table 1.

TABLE 1. Battery Energy storage system parameters.

Parameter	Value	Parameter	Value
W_{Bmax}	1000 kWh	ε_{SOC}	5%
$P_{B,min}$	–250 kW	$P_{B,max}$	250 kW
SOC_{min}	20%	SOC_{max}	95%
η_{ch}	90%	η_{dsc}	85%

C. MODEL VALIDATION

1) LOAD FLOW VALIDATION

In order to validate the load flow algorithm implemented by the authors, the MATPOWER toolbox has been used as reference, given its increased performance in power flow computation based on the Newton-Raphson method. The absolute errors between the authors' BFSM implementation and MATPOWER solver for bus voltages (in p.u.) and branch active powers (in kW) are presented in Fig. 11. The voltage errors are plotted with blue against the left Y-axis, while the power errors are plotted in orange against the right Y-axis. The dots represent the absolute error at each bus/branch and the dotted lines represent the mean absolute errors (MAE) for the entire network.

Fig. 11 shows absolute errors less than 9×10^{-10} p.u. for the bus voltages and less than 3.9×10^{-5} kW for branch active powers, while the MAE values are 4.6×10^{-10} p.u. for voltage and 9.9×10^{-6} kW for active power. Furthermore, the errors in both active and reactive powers supplied by the slack bus are less than 1.8×10^{-5} kVA. The obtained results prove the high accuracy of the authors' load flow implementation of the BFSM.

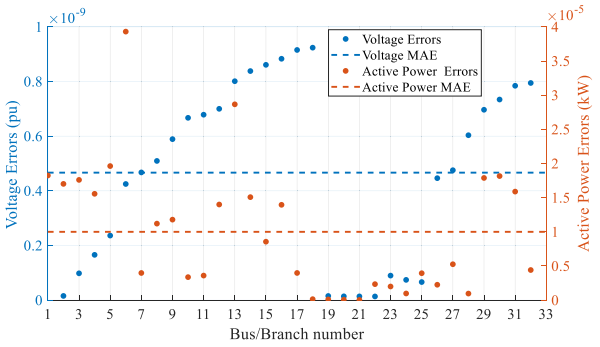

FIGURE 11. The absolute errors of the implemented load flow calculation model.

2) GPU VS. CPU COMPUTATIONAL SPEED COMPARISON

In order to compare the computational speed of the GPU parallel load flow calculation to the regular CPU series calculation, the following test is conducted: the number of load flow calculations necessary during one GWO iteration are performed using the same BFSM implemented on both GPU (actually a variant of the f_{obj}) and CPU. The test is carried out on the IEEE 33-bus test network, for a number of $N_{tot} = 10, 10^2, 10^3, 10^4$ individuals, meaning 240, 2400, 24000 and 240000 load flow calculations. The running time is measured for both GPU (t_{GPU}) and CPU (t_{CPU}) load flow implementations and the results are presented in Fig. 12, alongside with the ratio (R_t) between t_{CPU} and t_{GPU}. The numerical simulations have been run on a personal computer with an AMD Ryzen 7 2700 3.2GHz processor, 32GB RAM and a GeForce RTX^{TM}2060 graphics card.

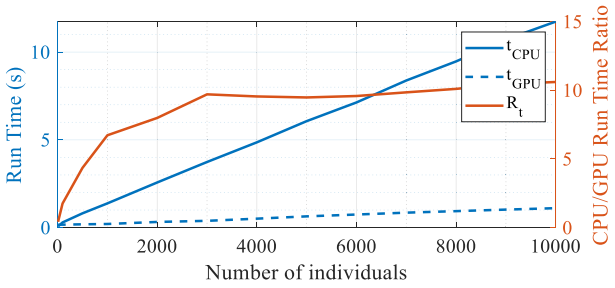

FIGURE 12. The running time variation for GPU and CPU load flow computation.

As Fig. 12 illustrates, when the number of individuals increases from 10^3 to 10^4, t_{CPU} increases 8.5 times from 1.38 s to 11.74 s, compared to t_{GPU} which increases 5.5 times from 0.2 s to 1.1 s. Thus, the test proves the superior computational speed of the GPU, which is more than 10 times faster compared to the CPU for 10^4 individuals and 6.7 times faster for 10^3 individuals.

3) MIGWO VS. GWO – PERFORMANCE ASSESSMENT

This section presents a comparison between MIGWO and GWO performance in minimizing several benchmark functions frequently utilized by researchers for testing

TABLE 2. MIGWO and GWO results for the benchmark functions.

Benchmark Function	Best		Average		Worst		Standard deviation	
	MIGWO	GWO	MIGWO	GWO	MIGWO	GWO	MIGWO	GWO
F1	**2.58E-16**	5.38E-16	**1.51E-15**	6.27E-15	**5.53E-15**	2.61E-14	**1.32E-15**	6.29E-15
F2	**8.27E-10**	1.59E-09	**2.15E-09**	4.64E-09	**4.83E-09**	1.20E-08	**1.04E-09**	2.17E-09
F3	**3.54E-06**	6.44E-06	**8.17E-05**	8.58E-05	**0.000224**	0.000612	**5.33E-05**	0.000115
F4	**3.19E-05**	5.91E-05	**9.19E-05**	0.00021	**0.000204**	0.000603	**4.68E-05**	0.000128
F5	**23.6282**	23.8962	**24.00885**	25.14377	**24.83031**	26.98881	**0.265231**	0.79298
F6	**0.000131**	0.000154	0.0002	**0.000187**	0.000254	**0.000217**	3.51E-05	**1.85E-05**
F7	**6.87E-06**	2.12E-05	5.72E-05	**5.26E-05**	0.000123	**9.95E-05**	2.63E-05	**2.30E-05**
F8	**-12560.7**	-8244.57	**-12536.4**	-7605.3	**-12369.6**	-6967.82	**52.6398**	371.8724
F9	**6.63E-08**	1.64E-07	**0.021166**	6.868876	**0.162309**	14.24403	**0.046863**	4.852734
F10	**4.91E-09**	6.52E-09	**8.70E-09**	2.00E-08	**1.57E-08**	3.67E-08	**3.06E-09**	8.91E-09
F11	**6.66E-16**	5.22E-15	**0.001972**	0.00403	**0.014777**	0.030288	**0.004312**	0.008259
F12	**3.82E-06**	4.12E-06	**7.59E-06**	0.000459	**1.07E-05**	0.006797	**1.50E-06**	0.001723
F13	8.49E-05	**7.48E-05**	**0.000116**	0.013108	**0.000159**	0.10102	**1.78E-05**	0.033751
F14	0.998004	0.998004	0.998004	0.998004	0.998004	0.998004	**6.24E-12**	1.41E-11
F15	0.000307	0.000307	0.000307	0.000307	0.000307	0.000307	**1.09E-09**	1.46E-09
F16	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	-1.03163	1.29E-09	**1.05E-09**
F17	0.397887	0.397887	0.397887	0.397888	0.397889	0.39789	**2.12E-07**	4.82E-07
F18	3	3	3	3	3	3	**2.42E-11**	9.74E-09
F19	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	-3.86278	**1.38E-07**	2.11E-07
F20	-3.32199	-3.32199	**-3.27839**	-3.23053	**-3.20309**	-3.19929	0.058272	**0.051321**
F21	**-10.1531**	-10.1526	**-9.30729**	-9.28906	**-5.05517**	-4.50617	**1.921957**	1.962855
F22	**-10.4029**	-10.4025	**-9.87335**	-9.86972	**-5.08764**	-5.08762	**1.613256**	1.621278
F23	**-10.5364**	-10.5359	**-10.357**	-10.3558	**-5.17558**	-5.17444	0.978607	**0.978598**

meta-heuristic algorithms performance. As the comparison is performed between MIGWO and GWO, this study has considered the 23 classical benchmark functions from the original GWO paper [14]: seven unimodal functions (F1 – F7), six multimodal functions (F8 – F13) and ten fixed-dimension multimodal functions (F14 – F23). Also, identical parameters and notations are used for these functions, therefore the detailed functions description is not provided in this paper. Both algorithms are applied for 30 consecutive times on each function, with identical parameters. The best, average and maximum values as well as the standard deviation for both algorithms are presented in Table 2.

In this study, the authors propose a new strategy, regarding population size and iterations number, that consists in employing a larger number of individuals, and fewer iterations. Firstly, by using parallel computing for the objective function evaluation, all the values are determined simultaneously, thus the computational time is comparable for 10^2, 10^3 and 10^4 individuals. Secondly, the iterations can only be performed by using series computation, meaning that the total computing time will increase linearly with the iterations number. Consequently, in order to reduce the computational time, the authors proposed a larger number of individuals ($N_{tot} = 10^4$) and a reduced number of iterations ($l_{max} = 100$), for both GWO and MIGWO algorithms. These parameters are further applied for the benchmark functions and the BESS scheduling problem as well.

The results provided in Table 2 reveal that the MIGWO outperforms the GWO algorithm on the majority of the benchmark functions. Given that all functions are minimization problems, lower values indicate better results of the optimization algorithms. The most notable improvement can be observed on F8 and F9 functions, where the average values obtained by the proposed MIGWO (-12536.4 and 0.0212) are

significantly lower than the GWO's (-7605.3 and 6.8689). The optimal values provided by the MIGWO are higher only for one function (F13). However, in this case, the other analyzed indicators (average, worst and standard deviation) are still better than GWO's. For six functions (F14 – F19) the best, average and worst results are approximately identical, with better results for the standard deviation obtained once again by MIGWO. Overall, the results show an improved exploration by the proposed algorithm, reflected on multi-modal functions average and worst values (as MIGWO avoids local minima points), without degrading the exploitation process, which can be observed in lower or comparable best values.

D. DAY-AHEAD SCHEDULING

The optimization model is applied in order to identify the day-ahead optimal scheduling of the BESS coordinated with the local RES generation. For this purpose, the load and generation profiles utilized by this paper are considered as day-ahead forecast data. Two different days are selected in this regard, and the results are obtained by running the MIGWO algorithm with $N_{tot} = 10^4$ individuals, for $l_{max} = 100$ iterations and selecting the best result achieved after five consecutive runs.

The first day-ahead optimal BESS scheduling is performed for 24-Dec-2019, especially selected as it represents the day with the largest relative power losses reduction from the entire year. The load and generation profiles (expressed in percentage) for each type of load and DG are presented in Figs. 9 and 10 from Section B, while Fig. 13 displays the hourly profile of the active power supplied by the slack bus alongside the total load and total generation profiles (for the entire network).

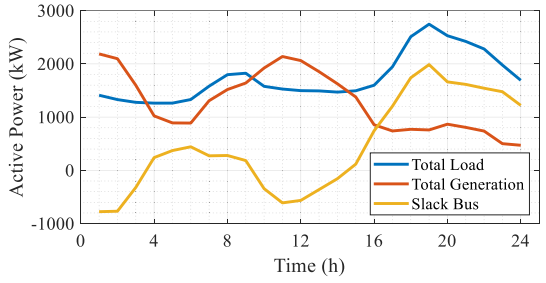

FIGURE 13. Hourly profiles of the active power supplied by the slack bus, total load and total generation for 24-Dec-2019.

The BESS optimal scheduling for 24-Dec-2019, considering $SOC_0 = 50\%$ is shown in Fig. 14: the blue bars plotted against the left Y-axis represent the BESS active power exchange (P_B in kW) and the orange line plotted against the right Y-axis is the SOC. The initial SOC corresponds on the graphic to time interval "0", and the other SOC values represent the BESS state of charge at the end of each interval. Note that the P_B value corresponding to time interval "4" shows the BESS behavior from 03.00 to 04.00.

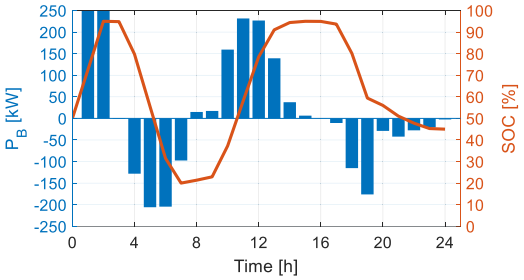

FIGURE 14. Battery power exchange and SOC variation in 24-Dec-2019 for $SOC_0 = 50\%$.

As it can be observed, the behavior of BESS is consistent with the local energy production and consumption, the battery charging taking place during the load valley (00.00-02.00), respectively during the high WT and PV generation (07.00-15.00), while its discharging takes place during the load peak (16.00-24.00) and during the night (03.00-07.00, to free capacity for storing high RES output during the day).

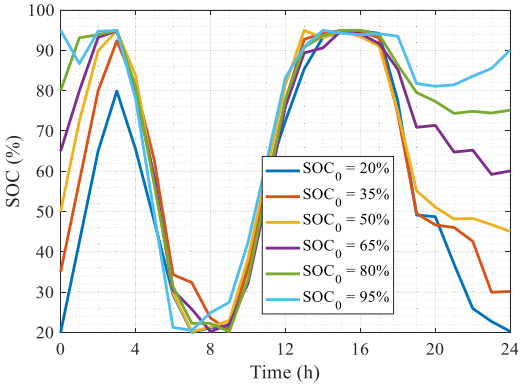

FIGURE 15. State-of-charge variation during 24-dec-2019 for different initial SOC values.

Considering the impact of the initial SOC value upon the daily battery performance, multiple SOC_0 values have been analyzed hereinafter. In this regard, the MIGWO algorithm was applied for six different SOC_0 values, namely: 20%, 35%, 50%, 65%, 80% and 95%. The resulting SOC variation during the day is represented in Fig. 15 for all six cases. A similar behavior can be observed in the daytime hours (from 06.00 to 21.00) for all six cases, while the SOC_0 influence is more relevant for the rest of the day. Also, the difference between the final and initial state of charge is less than 5% for all cases, which shows that GWO penalty functions correctly enforce the SOC-related constraints.

The active power losses (ΔP) expressed as percentage relative to the initial case (without the BESS) are given as a function of SOC_0 in Fig. 16.

Given the optimization model presented in this paper, the battery performance is evaluated in terms of active power losses reduction. Fig. 16 shows that the best battery performance is achieved for $SOC_0 = 20\%$, when ΔP are reduced to 88,1 % from the initial value. As the initial state of charge

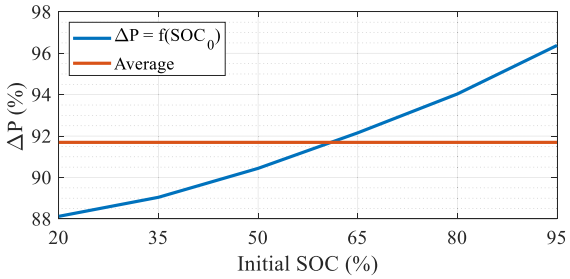

FIGURE 16. Active power losses obtained in 24-Dec-2019 for different initial SOC values.

value increases, the BESS has a lower impact on the active power losses reduction. This analysis reveals that the battery performance can be highly affected as the active power losses reduction is limited from 11.9% to 3.6%, when an inappropriate SOC_0 is chosen.

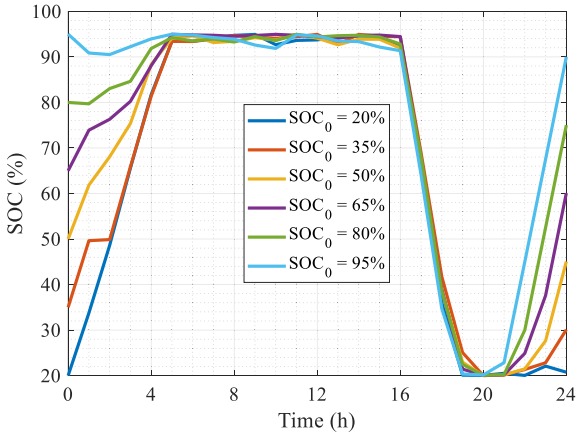

FIGURE 17. State-of-charge variation during 04-Jun-2019 for different initial SOC values.

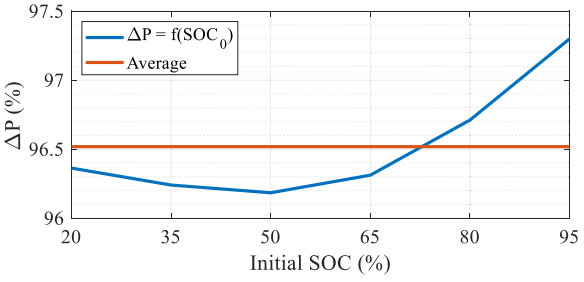

FIGURE 18. Active power losses obtained in 04-Jun-2019 for different initial SOC values.

The second day-ahead optimal BESS scheduling presented in this section is performed for 04-Jun-2019. As for the previous day, the MIGWO algorithm is applied for six different SOC_0 values and the resulting SOC variation during the day is presented in Fig. 17, while the active power losses for each SOC_0 value are shown in Fig. 18. The BESS behavior is similar to the previous case during the daytime hours, from 05.00 to 21.00 in order to reduce the active power losses, the difference occurring in rest of the day as the battery needs to comply with SOC-related constraints.

The results presented in Fig. 18 show that the best battery performance is achieved for $SOC_0 = 50\%$, when

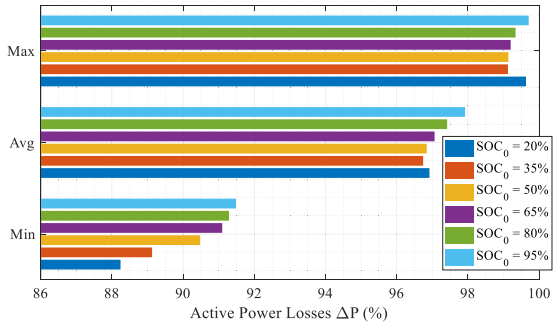

FIGURE 19. Daily minimum, average and maximum active power losses for every SOC_0 during the days of an entire year.

$\Delta P = 96.2\%$, while the power losses are increasing as the SOC_0 values move away from this value. In this case, the initial state of charge influence upon the battery performance is less important compared to the previous case, as power losses reduction is decreased from 3.8% when $SOC_0 = 50\%$ to 2.8% when $SOC_0 = 95\%$.

E. ONE-YEAR ANALYSIS

The battery operation is analyzed for a one-year period, by successively applying the BESS scheduling optimization model for each day of the year. In this regard, the input data are considered as historical data for one whole year and the daily scheduling is performed by applying the MIGWO algorithm with 700 individuals for 50 iterations. Depending on the number of iterations necessary for the load flow computation, the MIGWO requires from 9 up to 20 seconds per day, which adds up to a total running time between 60 and 90 minutes per year. Also, the analysis is performed for six different initial state of charge values: 20%, 35%, 50%, 65%, 80% and 95%.

One of the main concerns in performing analyzes based on large amounts of data is the computational time. By using the recommended number of individuals and iterations, discussed in section C, the current analysis would imply 52.56 billion load flow calculations, which would require 200 hours. In this case, a reduced loss of accuracy was tolerated in the study in order to keep the computational time to an acceptable range. By using the parameters mentioned above, the simulations imply 1.84 billion load flow calculations (24 hours profile × 700 individuals × 50 iterations × 365 days × 6 different SOC_0 values), leading to a total simulation time of only 7 hours (a 30 times reduction in computation time). Moreover, a significant curtailment of the simulation time is obtained due to the GPU-parallel load flow calculation, which is several times faster than the CPU load flow calculation, as previously shown.

The battery performance is evaluated for an entire year, for the six different SOC_0 values by comparing the active power losses – expressed as percentage relative to the initial case. For each one of the SOC_0 values, the one-year minimum, average and maximum are presented in Fig. 19.

The lowest one-year average ΔP value of 96.7% is obtained for $SOC_0 = 35\%$, while the 50% and 20% follow closely with 96.8% and 96.9%, respectively. Then, as the

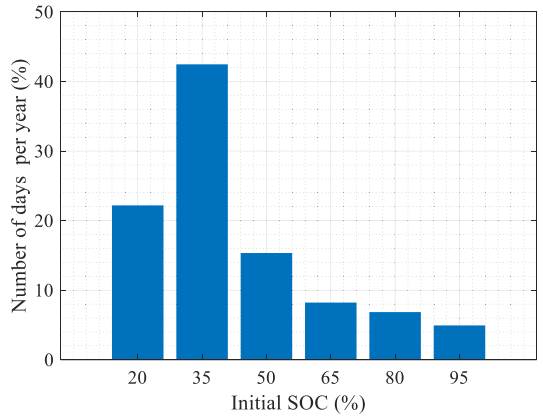

FIGURE 20. Number of days per year (in %) when each SOC_0 provides the lowest active power losses.

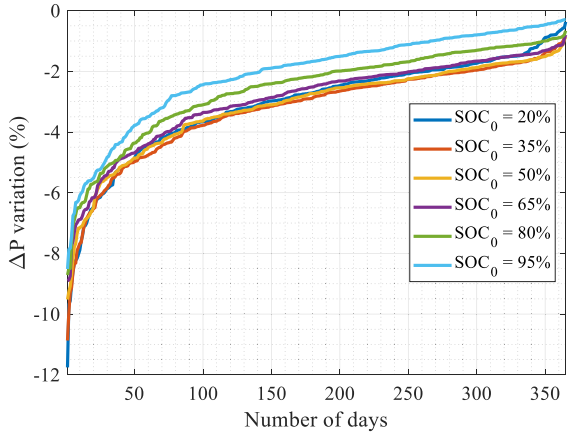

FIGURE 21. Power losses variation (in percentage) for one-year period.

initial SOC increases, the average ΔP value also increases up to 97.9%. The minimum one-year active power losses of 88.1% are obtained for $SOC_0 = 20\%$, and the values increase as the initial SOC increases. The lowest maximum one-year ΔP are obtained for 35% and 50% initial SOC, while the highest values are represented by the extreme SOC_0 – 20% and 95%. These results suggest that an initial SOC value of 35% will obtain the lowest average active power losses throughout the entire year.

Fig. 20 presents the number of days in which every SOC_0 is considered the recommended, as the lowest active power losses are obtained, in comparison to all the other SOC_0 values. The number of days is expressed as percentage relative to the total number of days within a year. The SOC_0 value of 35% assures the best performance in 42.5% cases throughout the year, while the higher SOC_0 values of 65%, 80% and 95% obtain the lowest active power losses in less than 10% of the days, each. By considering both analyses presented in Fig. 19 and Fig. 20, the best battery performance is obtained by using the initial state of charge value of 35%, which will provide the best power losses reduction and the highest number of days when this minimization occurs.

Finally, the active power losses variation – expressed in percentage relative to the initial case – are represented, in ascending order for the six SOC_0 values, in Fig. 21. The results reveal that the active power losses are reduced by at least 5% in a number of 49, 46, 45, 36, 35 and 28 days for SOC_0 values of 35%, 20%, 50%, 65%, 80% and 95% respectively, and by at least 2% in a number of 293, 280, 264, 252, 196 and 141 days for SOC_0 values of 35%, 50%, 20%, 65%, 80% and 95% respectively. The results reveal as well that the recommended initial SOC value for the entire year is 35%. Also 20%, 50% and 65% are relatively good SOC_0 values, while the most inappropriate choices are 80% and 95%.

V. CONCLUSION

In this paper, the short-term operation scheduling problem of a BESS has been solved, by the optimal coordination of the storage system with the local available RES and the energy demand in an active distribution network, focusing on the total power losses minimization. In this regard, the authors propose a novel Mutation-Improved GWO algorithm to overcome the risk of convergence in local optimums of the standard technique. 23 benchmark functions were used in order to assess the performance of the proposed algorithm compared to the original GWO, the increased efficiency of MIGWO being noticed in most cases. The study also exploits the benefits of parallel GPU accelerated global search, in order to explore the large and complex search space of the optimal BESS scheduling problem for better solutions. A modified IEEE 33-bus system has been analyzed, using forecasted data for RES generation and typical consumption profiles for various types of loads. For a more efficient reduction of the power losses occurred in an ADN, the importance of choosing the right initial state of charge for the storage system was reflected in the obtained results.

The time-consuming process of multiple power flow computation required in meta-heuristics algorithms can be shortened using simultaneous (parallel) computing and clustering techniques. Therefore, future work focuses on solving large-scale problems in power systems implying numerous controllable devices, such as storage, distributed generation and FACTS, by implementing GPU-cluster computation.

REFERENCES

[1] Y. Yang, W. Zhang, J. Jiang, M. Huang, and L. Niu, "Optimal scheduling of a battery energy storage system with electric vehicles' auxiliary for a distribution network with renewable energy integration," *Energies*, vol. 8, no. 10, pp. 10718–10735, 2015.
[2] L. Petrichenko, R. Varfolomejeva, A. Gavrilovs, A. Sauhats, and R. Petricenko, "Evaluation of battery energy storage systems in distribution grid," in *Proc. IEEE Int. Conf. Environ. Electr. Eng. IEEE Ind. Commercial Power Syst. Eur. (EEEIC/I&CPS Europe)*, Palermo, Italy, Jun. 2018, pp. 1–6.
[3] A. A. Hadi, C. A. S. Silva, E. Hossain, and R. Challoo, "Algorithm for demand response to maximize the penetration of renewable energy," *IEEE Access*, vol. 8, pp. 55279–55288, 2020.
[4] I. Konstantelos and G. Strbac, "Capacity value of energy storage in distribution networks," *J. Energy Storage*, vol. 18, pp. 389–401, Aug. 2018.
[5] W. Liu, S. Niu, and H. Xu, "Optimal planning of battery energy storage considering reliability benefit and operation strategy in active distribution system," *J. Mod. Power Syst. Clean Energy*, vol. 5, no. 2, pp. 177–186, Mar. 2017.
[6] N. Karthikeyan, B. R. Pokhrel, J. R. Pillai, and B. Bak-Jensen, "Utilization of battery storage for flexible power management in active distribution networks," in *Proc. IEEE Power Energy Soc. Gen. Meeting (PESGM)*, Portland, OR, USA, Aug. 2018, pp. 1–5.

[7] M. Khalid, U. Akram, and S. Shafiq, "Optimal planning of multiple distributed generating units and storage in active distribution networks," *IEEE Access*, vol. 6, pp. 55234–55244, 2018.

[8] B. Xu, Y. Shi, D. S. Kirschen, and B. Zhang, "Optimal battery participation in frequency regulation markets," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6715–6725, Nov. 2018.

[9] J. W. Shim, G. Verbic, K. An, J. H. Lee, and K. Hur, "Decentralized operation of multiple energy storage systems: SOC management for frequency regulation," in *Proc. IEEE Int. Conf. Power Syst. Technol. (POWERCON)*, Wollongong, NSW, Australia, Sep. 2016, pp. 1–5.

[10] L. Toma, M. Sanduleac, S. A. Baltac, F. Arrigo, A. Mazza, E. Bompard, A. Musa, and A. Monti, "On the virtual inertia provision by BESS in low inertia power systems," in *Proc. IEEE Int. Energy Conf. (ENERGYCON)*, Limassol, Cyprus, Jun. 2018, pp. 1–6.

[11] A. Hussain, V.-H. Bui, and H.-M. Kim, "A proactive and survivability-constrained operation strategy for enhancing resilience of microgrids using energy storage system," *IEEE Access*, vol. 6, pp. 75495–75507, 2018.

[12] N. D. Laws, K. Anderson, N. A. DiOrio, X. Li, and J. McLaren, "Impacts of valuing resilience on cost-optimal PV and storage systems for commercial buildings," *Renew. Energy*, vol. 127, pp. 896–909, Nov. 2018.

[13] S. Tsianikas, J. Zhou, N. Yousefi, and D. W. Coit, "Battery selection for optimal grid-outage resilient photovoltaic and battery systems," pp. 10–15, 2019, *arXiv:1901.11389*. [Online]. Available: https://arxiv.org/abs/1901.11389

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[15] M. Rambabu, G. V. Nagesh Kumar, and S. Sivanagaraju, "Optimal power flow of integrated renewable energy system using a thyristor controlled SeriesCompensator and a grey-wolf algorithm," *Energies*, vol. 12, no. 11, p. 2215, Jun. 2019.

[16] Z. G. Hassan, M. Ezzat, and A. Y. Abdelaziz, "Enhancement of power system operation using grey wolf optimization algorithm," in *Proc. 19th Int. Middle East Power Syst. Conf. (MEPCON)*, Dec. 2017, pp. 392–402.

[17] R. Jamal, B. Men, and N. H. Khan, "A novel nature inspired meta-heuristic optimization approach of GWO optimizer for optimal reactive power dispatch problems," *IEEE Access*, vol. 8, pp. 202596–202610, 2020.

[18] I. B. M. Taha and E. E. Elattar, "Optimal reactive power resources sizing for power system operations enhancement based on improved grey wolf optimiser," *IET Gener., Transmiss. Distrib.*, vol. 12, no. 14, pp. 3421–3434, Aug. 2018.

[19] K. S. EI-Bidairi, H. Duc Nguyen, S. D. G. Jayasinghe, and T. S. Mahmoud, "Multiobjective intelligent energy management optimization for grid-connected microgrids," in *Proc. IEEE Int. Conf. Environ. Electr. Eng. IEEE Ind. Commercial Power Syst. Eur. (EEEIC/I&CPS Europe)*, Palermo, Italy, Jun. 2018, pp. 1–6.

[20] A. Khlifi, S. Chettih, and B. Bentouati, "A new hybrid algorithm of particle swarm optimizer with grey wolves' optimizer for solving optimal power flow problem," *Leonardo Electron. J. Pract. Technol.*, no. 32, pp. 249–270, Jan. 2018.

[21] I. Araújo, V. Tadaiesky, D. Cardoso, Y. Fukuyama and Á. Santana, "Simultaneous parallel power flow calculations using hybrid CPU-GPU approach," *Int. J. Elect. Power Energy Syst.*, vol. 105, pp. 229–236, Feb. 2019.

[22] J. R. Cheng and M. Gen, "Accelerating genetic algorithms with GPU computing: A selective overview," *Comput. Ind. Eng.*, vol. 128, pp. 514–525, Feb. 2019.

[23] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018.

[24] W. Long and S. Xu, "A novel grey wolf optimizer for global optimization problems," in *Proc. IEEE Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Xi'an, China, Oct. 2016, pp. 3–5.

[25] P. P. Biswas and P. N. Suganthan, "Large initial population and neighborhood search incorporated in LSHADE to solve CEC2020 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Glasgow, U.K., Jul. 2020, pp. 19–24.

[26] M. A. Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowl.-Based Syst.*, vol. 172, pp. 42–63, May 2019.

[27] E. K. Burke, S. Gustafson, and G. Kendall, "Diversity in genetic programming: An analysis of measures and correlation with fitness," *IEEE Trans. Evol. Comput.*, vol. 8, no. 1, pp. 47–62, Feb. 2004.

[28] H. Maaranen, K. Miettinen, and M. M. Mäkelä, "Quasi-random initial population for genetic algorithms," *Comput. Math. Appl.*, vol. 47, no. 12, pp. 1885–1895, Jun. 2004.

[29] P. A. Diaz-Gomez and D. F. Hougen, "Initial population for genetic algorithms: A metric approach," in *Proc. Int. Conf. Genetic Evol. Methods, GEM*, Las Vegas, NV, USA, Jun. 2007, pp. 43–49.

[30] J.-S. Wang and S.-X. Li, "An improved grey wolf optimizer based on differential evolution and elimination mechanism," *Sci. Rep.*, vol. 9, no. 1, p. 7181, Dec. 2019.

[31] J. A. M. Rupa and S. Ganesh, "Power flow analysis for radial distribution system using backward/forward sweep method," *World Acad. Sci., Eng. Technol. Int. J. Elect. Comput. Eng.*, vol. 8, no. 10, pp. 1621–1625, 2014.

[32] S. Sunisith and K. Meena, "Backward/forward sweep based distribution load flow method," *Int. Elect. Eng. J. (IEEJ)*, vol. 5, no. 9, pp. 1539–1544, 2014.

[33] MathWorks. (2020). *Parallel Computing Toolbox User's Guide*. Accessed: May 30, 2020. [Online]. Available: https://www.mathworks.com/help/pdf_doc/parallel-computing/parallel-computing.pdf

[34] M. E. Baran and F. F. Wu, "Network reconfiguration in distribution systems for loss reduction and load balancing," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1401–1407, Apr. 1989.

[35] *Simulated Load Profiles for DOE Commercial Reference Buildings (17 Years Using NSRD Data), U.S. Department of Energy*. Accessed: Jan. 15, 2020. [Online]. Available: https://openei.org/datasets/dataset/simulated-load-profiles-17year-doe-commercial-reference-buildings

[36] S. Pfenninger and I. Staffell, "Long-term patterns of European PV output using 30 years of validated hourly reanalysis and satellite data," *Energy*, vol. 114, pp. 1251–1265, Nov. 2016.

[37] I. Staffell and S. Pfenninger, "Using bias-corrected reanalysis to simulate current and future wind power output," *Energy*, vol. 114, pp. 1224–1239, Nov. 2016.

[38] S. Pfenninger and I. Staffell. *Renewables.Ninja*. Accessed: Jan. 15, 2020. [Online]. Available: https://www.renewables.ninja/

[39] P. Nikolaidis and A. Poullikkas, "Cost metrics of electrical energy storage technologies in potential power system operations," *Sustain. Energy Technol. Assessments*, vol. 25, pp. 43–59, Feb. 2018.

DORIAN O. SIDEA (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in power systems engineering from the University "Politehnica" of Bucharest, Romania, in 2012, 2014, and 2018, respectively. He is with the Electric Power Systems Department, Power Engineering Faculty, University "Politehnica" of Bucharest, since 2014, and currently serves as a Lecturer. His research interests include power systems analysis, metaheuristic algorithms, and artificial neural networks applications in power systems, energy storage systems, and microgrids.

IRINA I. PICIOROAGA (Student Member, IEEE) was born in Galati, Romania. She received the B.S. and M.S. degrees in power systems engineering from the University "Politehnica" of Bucharest in 2014 and 2016, respectively, where she is currently pursuing the Ph.D. degree in power systems engineering. She has also served as a Teaching and Research Assistant with the Department of Electrical Power Systems, University "Politehnica" of Bucharest, since 2016. Her research interests include the optimal operation of distribution networks, distributed energy generation, energy storage systems, and microgrids.

CONSTANTIN BULAC (Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from the University "Politehnica" of Bucharest, in 1982 and 1998, respectively. He is currently a Professor and a Ph.D. supervisor with the Department of Electrical Power Systems, University "Politehnica" of Bucharest. His research interests include advanced solutions in power systems, stability assessment, and AI applications in power systems. He has authored/coauthored several books and multiple papers in the field of control and operation power systems, FACTS devices, and artificial intelligence.