Security Analysis and Optimization: A Hybrid Multiswarm Particle Swarm Optimizer

Venkata Ramana Gupta Nallagattla, Venugopal Gaddam, S Ravi Kishan, Tirumalasetti Lakshmi Narayana, N Venkateswara Rao, S N V Jyotsna Devi Kosuru

***Abstract*—SCOPF ensures power system reliability while reducing operational costs and losses. Traditional approaches struggle with SCOPF problems' nonlinearity and nonconvexity, requiring improved optimization. This work proposes an Adaptive Swarm Hybrid Optimizer (ASHO) to solve SCOPF problems. The ASHO balances exploration and exploitation, avoids premature convergence, and increases solution variety using multi-swarm dynamics, adaptive inertia weights, and mutation operators. The proposed method is tested on the IEEE 30-bus system. Simulations show that ASHO outperforms PSO, GA, and DE. ASHO minimizes generation costs, transmission losses, and voltage stability while meeting operational restrictions. Compared to other approaches, it reduces generation costs by 1.5% and improves voltage profiles and convergence. ASHO also adheres to system restrictions during significant outages, assuring security. These results show that ASHO can provide efficient and dependable SCOPF solutions in current power systems. This approach will be extended to dynamic SCOPF and larger test systems in future study.**

***Index Terms*—Security-Constrained Optimal Power Flow, Hybrid Multiswarm Particle Swarm Optimizer, power system optimization, swarm intelligence.**

I. Introduction

THE increasing complexity of power systems, along with the rising demand for reliable and sustainable energy, requires effective operational planning and optimization. Security-Constrained Optimal Power Flow (SCOPF) has become an essential instrument in this domain [1]. This approach enhances the traditional Optimal Power Flow (OPF) problem by integrating security constraints, thereby ensuring the reliability and robustness of power systems during contingency scenarios, including line outages or generator failures. The main goals of SCOPF include minimizing generation costs, reducing transmission losses, and ensuring secure and stable system operation while adhering to all system constraints. Solving SCOPF presents considerable challenges owing to its nonlinear, nonconvex, and high-dimensional characteristics [2]. The Optimal Power Flow (OPF) problem, introduced in the 1960s, seeks to identify the optimal generation dispatch that minimizes costs or losses while adhering to power balance, generator limits, voltage constraints, and transmission line limits. SCOPF enhances its framework by integrating contingencies, thereby providing a more comprehensive approach for real-world applications [3], [4]. The integration of renewable energy sources and the growing complexity of contemporary power grids have rendered SCOPF a vital research domain for maintaining efficient and secure grid operations. Conventional optimization techniques, including linear programming (LP), quadratic programming (QP), and nonlinear programming (NLP), have been widely utilized to address OPF and SCOPF issues. These methods are frequently constrained by their inability to address nonconvexity and discrete variables, which are intrinsic to SCOPF formulations [5], [6]. The computational burden of these methods escalates exponentially with the size and complexity of the power system, rendering them less appropriate for large-scale systems or real-time applications.

A. *The Role of Metaheuristic Algorithms in SCOPF*

When it comes to solving SCOPF difficulties, metaheuristic algorithms have gained momentum as a solution thanks to their ability to overcome the constraints of older approaches. These algorithms, which are derived from natural processes such as evolution and the behavior of swarms, provide foundations that are both resilient and adaptable, making them suitable for handling difficult optimization issues. Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Differential Evolution (DE) are three of these concepts that have received a significant amount of research. In particular, PSO has been praised for its inherent simplicity, the ease with which it may be implemented, and its capacity to deal with nonconvex issues [7], [8]. The typical PSO algorithm, on the other hand, frequently experiences problems with early convergence and

Manuscript received March 27, 2025; revised July 4, 2025.

Venkata Ramana Gupta Nallagattla is an Assistant Professor of Computer Science and Engineering Department, School of Computing, Amrita Vishwa Vidyapeetham, Amaravati Campus, Amaravati, Andhra Pradesh, India (e-mail: nallagattla@gmail.com).

Venugopal Gaddam is an Associate Professor of Computer Science and Engineering AI & ML Department, B V Raju Institute of Technology, Narsapur, Medak, Hyderabad, Telangana, India (e-mail: venugopal.gaddam@gmail.com).

S Ravi Kishan is an Associate Professor of Computer Science and Engineering Department, Siddhartha Academy of Higher Education Deemed to be University, Kanuru, Vijayawada, Andhra Pradesh, India (e-mail: suraki@vrsiddhartha.ac.in).

Tirumalasetti Lakshmi Narayana is an Assistant Professor of Electrical and Electronics Engineering Department, Aditya University, Surampalem, Andhra Pradesh, India (e-mail: tlaxman17@gmail.com).

N Venkateswara Rao is a Professor of Computer Science and Engineering AI & ML Department, RVR & JC College of Engineering, Chowdavaram, Guntur, Andhra Pradesh, India (e-mail: vnaramala@gmail.com).

S N V Jyotsna Devi Kosuru is an Assistant Professor of Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India (e-mail: jyotsnakosuru@gmail.com).

stagnation in local optimal solutions, particularly in high-dimensional and multimodal situations such as SCOPF.

In order to overcome the limitations of the traditional Particle Swarm Optimizer (PSO) when it comes to handling SCOPF problems, this work presents a Hybrid Multiswarm Particle Swarm Optimizer (HMPSO). The HMPSO that has been proposed incorporates a number of major advances, including the following: HMPSO makes use of many interacting swarms, each of which explores various regions of the solution space. Because of this, the algorithm is better able to explore and break out of local optimum situations. The inertia weight is dynamically adjusted by the algorithm in order to strike a balance between exploration and exploitation while the optimization process is being carried out [9]. Additionally, a mutation process is introduced in order to prevent premature convergence and to introduce diversity into the universe of possible solutions. Validation of the proposed method is performed on the IEEE 30-bus system, which is a standard benchmark in the field of power system studies. When compared to regular PSO, GA, and DE, HMPSO is shown to be superior in terms of cost minimization, loss reduction, constraint satisfaction, and computing efficiency. This is demonstrated by comparative analysis.

II. Related Work

Methods that were used in the past to solve OPF and SCOPF focused on deterministic optimization techniques, such as LP, QP, and NLP. was one of the first people to formulate the OPF issue as a mathematical programming assignment. Following this, subsequent developments brought about the introduction of SCOPF formulations that included N-1 contingency requirements. Despite the fact that these approaches were able to produce accurate solutions under specific circumstances, their usefulness was restricted due to the complexity of the computations involved and the sensitivity to the initial conditions [10]. The implementation of interior-point methods and decomposition techniques was done with the intention of enhancing the efficiency of computation. An example of this would be a gradient-based technique for OPF, while other studies investigated the use of Lagrangian relaxation for SCOPF applications. Even with all of these efforts, the performance of classical approaches frequently deteriorated when nonconvexity, discrete variables, and large-scale systems were present [11]. The SCOPF solution methods are capable of doing massive computations while simultaneously improving tractability and hence preserving accuracy. The problem of SCOPF has been solved in a variety of ways, and several approaches to addressing it have been proposed. These approaches have been thoroughly investigated. The authors of reference [12] presented a method known as dynamic multichain particle swarm optimization (DMCPSO) in order to solve the SCOPF problem. Both a dynamic multichain architecture and an adaptive parameter regulation mechanism are utilized in its operation. It is advantageous to work toward minimizing the planned cost while taking into account the needs for system capacity and the limits imposed by operational security. An approach was presented by the authors in reference [13] in order to handle the preventative SCOPF (PSCOPF) problem for the purpose of implementation in power system planning and operation. The N-1-1 contingency analysis was completed with a high level of competence. The implementation of the reactive compensation approach, which was designed to alleviate post-contingency voltage concerns, was completely effective. The technique described in reference [14] provides specifics regarding how to evaluate the SCOPF solution while taking into account the probabilistic generation and transmission contingencies. By maximizing the amount of money spent on security, this was able to successfully achieve the recommended degree of system security. The authors of reference [15] provided a solution to address the SCOPF problem for a hybrid AC/DC grid. That method was described in the reference. This is applicable for preventative SCOPF in which corrective steps are forbidden after the contingency has occurred, as well as for corrective SCOPF with control action restrictions that can be modified. An approach known as hybrid multiswarm particle swarm optimization (HMPSO) was developed by the authors of reference [16] in order to solve the SCOPF problem. It is effective to minimize the predefined cost while taking into consideration the requirements for system capacity and the constraints on operational security. The adaptive partitioning flower pollination method was utilized by the authors of reference [17] in order to solve the SCOPF problem that was present in a utility grid. The reference [18] provides a full analysis of the key challenges and probable trends that are associated with SCOPF computations.

For the purpose of addressing the SCOPF problem, which involves a large number of line outages, a constraint-driven machine learning (ML) solution has been developed, as mentioned in number 19. By utilizing line outage distribution factors, often known as LODF, this objective can be successfully realized. Deterministic security and probabilistic security are both evaluated using this method, which is a deterministic method. A comprehensive investigation into machine learning proximity-based approaches for the use of SCOPF solutions was carried out by the authors of reference [20]. When comparing the effectiveness of the various techniques, several criteria, including as load distribution, power factors, online generators, network topology, and generator costs, are taken into consideration throughout the evaluation process. A strategy known as a dynamic fitness-distance balance-based growth optimizer (dFDB-GO) is described in reference [21] for the purpose of solving SCOPF in utility transmission networks. The SCOPF issue was successfully resolved with the help of this technique, which achieved a mean success rate of 94.87% through its implementation. A methodology was presented by the authors in reference [22] to address the integrated ac-dc SCOPF problem for substantial power systems. The authors also implemented the methodology in the Australian National Electricity Market. In situations that occur in real time, this method is able to effectively produce a solution in just five minutes. The authors of reference [22] presented a mathematical programming solution with the intention of tackling SCOPF while simultaneously adding dynamic security restrictions within an AC-microgrid. This method is beneficial for both transitioning to an island position and operating on an island.Metaheuristic algorithms have demonstrated their potential, yet there are still many obstacles to overcome. The standard PSO algorithm has a tendency to converge too quickly, and its performance is strongly dependent on the parameters that are tuned. Hybrid techniques, despite their effectiveness, frequently result in an increase in the complexity of algorithms and the amount of requisite processing. Additionally, the majority of research concentrate on static SCOPF scenarios, with only a limited amount of consideration given to dynamic and real-time applications [24], [25].

This study aims to address these limitations by developing a robust and efficient HMPSO for SCOPF. The specific objectives are to:

Design a hybrid multiswarm framework that enhances exploration and prevents premature convergence.

Incorporate adaptive inertia weights and mutation operators to improve solution diversity and convergence.

Evaluate the proposed HMPSO on the IEEE 30-bus system, with comparisons to standard PSO, GA, and DE.

Assess the robustness of HMPSO under contingency scenarios to demonstrate its applicability in real-world power systems.

III. SCOPF Problem Formulation

This section describes the mathematical formulation of the Security-Constrained Optimal Power Flow (SCOPF) problem. The objective function of the optimal power flow problem is to minimize the total generator fuel cost and expressed as follows:

$$J = \sum_{i=1}^{NG} (a_i + b_i P_{Gi} + c_i P_{Gi}^2) \quad (1)$$

The OPF problem is subjected to the following equality and inequality constraints.

Equality Constraints: These are the set of power flow equations that govern the power system and expressed as follows:

$$P_{Gi} - P_{Di} = \sum_{j=1}^{NB} |V_i||V_j||Y_{ij}| \cos(\theta_{ij} - \delta_i + \delta_j) \quad (2)$$

$$Q_{Gi} - Q_{Di} = -\sum_{j=1}^{NB} |V_i||V_j||Y_{ij}| \sin(\theta_{ij} - \delta_i + \delta_j) \quad (3)$$

Inequality Constraints: These are the set of constraints that represent the power system operational limits and security limits.

(i) Generation constraints:

Generator voltage, real power generation and reactive power generation are constrained as follows:

$$V_{Gi}^{\min} \le V_{Gi} \le V_{Gi}^{\max} \qquad i \in \text{NG} \quad (4)$$

$$P_{Gi}^{\min} \le P_{Gi} \le P_{Gi}^{\max} \qquad i \in \text{NG} \quad (5)$$

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max} \qquad i \in \text{NG} \quad (6)$$

(ii) Transformer constraints:

Transformer tap settings are constrained as follows:

$$T_i^{\min} \le T_i \le T_i^{\max} \qquad i \in \text{NT} \quad (7)$$

(iii) Security constraints:

The voltage at load buses and transmission line loadings are constrained as follows:

$$V_{Li}^{\min} \le V_{Li} \le V_{Li}^{\max} \qquad i \in \text{NLB} \quad (8)$$

$$S_{Li} \le S_{Li}^{\max} \qquad i \in \text{NL} \quad (9)$$

Where

a_i, b_i and c_i Cost coefficients of the i^{th} generator, \$/h, \$/MWh and \$/(MW)^{2}h respectively

P_{Gi} Total real power generation at bus i (MW)

Q_{Gi} Total reactive power generation at bus i (MVar)

P_{Di} Total real power load demand at bus i (MW)

Q_{Di} Total reactive power load demand at bus i (MVar)

V_i Voltage magnitude at bus i

V_j Voltage magnitude at bus j

δ_i Voltage phase angle at bus i

δ_j Voltage phase angle at bus j

$|Y_{ij}|$ Magnitude of ij^{th} element of bus admittance matrix

θ_{ij} Angle of ij^{th} element of bus admittance matrix

T_i Tap setting of transformer i

S_{Li} Transmission line loading at line i (MVA)

NB Total number of buses

NLB Total number of load buses

NG Total number of generators

NL Total number of transmission lines

NT Total number of regulating transformers

IV. Hybrid and Modified Particle Swarm Optimization (HMPSO)

The proposed HMPSO algorithm improves upon the standard PSO by incorporating multiswarm dynamics, adaptive inertia weights, and a mutation operator. Particle Swarm Optimization (PSO) is an optimization algorithm that utilizes a population-based approach, drawing inspiration from the collective behavior observed in bird flocks or schools of fish. James Kennedy and Russell Eberhart first introduced it in 1995. PSO is commonly utilized across multiple domains because of its straightforward nature, ease of application, and effectiveness in addressing optimization challenges.

Hybrid and Modified Particle Swarm Optimization (HMPSO) denotes advanced iterations of the conventional PSO algorithm. These iterations integrate hybridization with various optimization methods, alongside adjustments in the algorithm's framework, parameters, or operators. The objective is to tackle the limitations of conventional PSO, including premature convergence, insufficient diversity, and challenges in managing complex, high-dimensional issues. Hybridization integrates PSO with various optimization techniques to improve performance by utilizing the advantages of different approaches. a) Genetic Algorithm (GA): In the Hybrid PSO-GA framework, operators such as crossover and mutation are employed to enhance diversity and prevent premature convergence. Integrates the states of two particles to generate progeny. Translates arbitrary alterations in particle locations to investigate novel areas. b) Differential Evolution (DE): The mutation and crossover strategies of DE are integrated to improve global exploration. c) Simulated Annealing (SA): The temperature-based exploration mechanism of SA is utilized in HMPSO to improve local search capabilities. The probability of accepting a suboptimal solution is provided.

HMPSO can incorporate ES operators such as selection and recombination to improve convergence and robustness.

Algorithm Steps

Initialization:Randomly initialize the positions and velocities of all particles within predefined bounds. Evaluate the fitness of each particle's initial position.

Personal Best (p_{best}): Each particle remembers its best-known position, where it achieved the highest fitness.

Global Best (g_{best}): The best-known position achieved by any particle in the swarm.

Velocity Update: The velocity of each particle is updated using the formula:

$$V_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1 \cdot (p_{best} - x_i(t)) + c_2 \cdot r_2 \cdot (g_{best} - x_i(t)) \quad (10)$$

$v_i(t+1)$: Updated velocity of particle ii.

xi(t): Current position of particle ii.

w: Inertia weight, balancing exploration and exploitation.

c_1, c_2: Acceleration coefficients (typically between 0 and 2).

r_1, r_2: Random values between 0 and 1.

Position Update: The new position is calculated as:

$$x_i(t+1) = x_i(t) + vi(t+1) \quad (11)$$

Evaluation: Evaluate the new position using the fitness function. Update p_{best} and g_{best} if better solutions are found.

Termination: The process continues iteratively until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory fitness value. This methodology provides a clear framework for implementing and validating the HMPSO algorithm in the context of SCOPF. Let me know if you need further refinements or additional details.

V. Results and Discussion

The findings of implementing the suggested Hybrid Multiswarm Particle Swarm Optimizer (HMPSO) to figure out how to solve the Security-Constrained Optimal Power Flow (SCOPF) problem for the IEEE 30-bus system are shown in this part. A convergence behavior analysis, a cost minimization analysis, and a computational efficiency analysis are performed on the observed results. Standard PSO (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE) are all used as benchmarks for comparison. The IEEE 30-bus test system is comprised of 21 load buses, six generators, and 41 transmission lines all working together. At the base load, there are 283.4 MW and 126.2 MVAr. as determined by the IEEE standard dataset, generation cost coefficients and system characteristics are presented here. There are fifty particles in each swarm of the HMPSO and SPSO population. In HMPSO, there are three swarms in total. The maximum number of iterations is one hundred. The penalty factor is 1000, while the mutation factor is 0.1.

The overall generation costs for several optimization methods (ASHO, SPSO, DE, and GA) are compared in Figure 1, which highlights the comparison. It reveals that ASHO is able to achieve the lowest generating cost, which is $802.34, indicating the effectiveness of the company in their efforts to minimize operational expenses. In Figure 2, the performance of the Hybrid Multiswarm Particle Swarm Optimizer (HMPSO) is displayed in comparison to the performance of various optimization approaches (PSO, GA, and DE) when it comes to solving the Security-Constrained Optimal Power Flow (SCOPF) problem: The generation cost decrease achieved by HMPSO is the largest of any method, coming in at 1.5%, which is a substantial improvement by comparison to other approaches. Transmission Loss Reduction: HMPSO displays the most efficiency in reducing transmission losses, reaching a reduction of 6.3% and demonstrating the maximum efficiency. When compared to other methods, HMPSO is the most efficient in terms of convergence, as it requires just 70 iterations to get a full convergence.

When it comes to optimizing energy generation, cost minimization is an increasingly important component. While simultaneously satisfying the demand and other operational restrictions, the objective is to distribute the available resources in such a way as to reduce the overall cost of generation as much as possible. As part of this investigation, we investigate the generation schedules that result from the application of four distinct optimization strategies: HMPSO, SPSO, GA, and DE. The scenario known as the Base Case, which does not include any contingencies, is assessed in order to determine which approach results in the lowest cost.

A. Base Case (No Contingency)

HMPSO, which stands for hybrid multi-particle swarm optimization, is the optimization technique that produces the lowest total generation cost when compared to the other techniques that are taken into consideration. The primary purpose is to distribute the generation of power among the various producers in a manner that minimizes costs while simultaneously satisfying the level of demand that is required. Standard Particle Swarm Optimization (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE) are the other optimization strategies that are being studied. The method of optimization takes into account a number of different aspects, including the operating limits of each generator, the cost curves, and the generating capacity offered by each generator.

The purpose of these methods is to distribute the load evenly among the many generators while simultaneously reducing the amount of fuel used and the operational inefficiencies that occur. Table 1 provides a summary of the optimized generation schedule for each optimization strategy using the Base Case scenario, along with the expenditures that are associated with each technique.

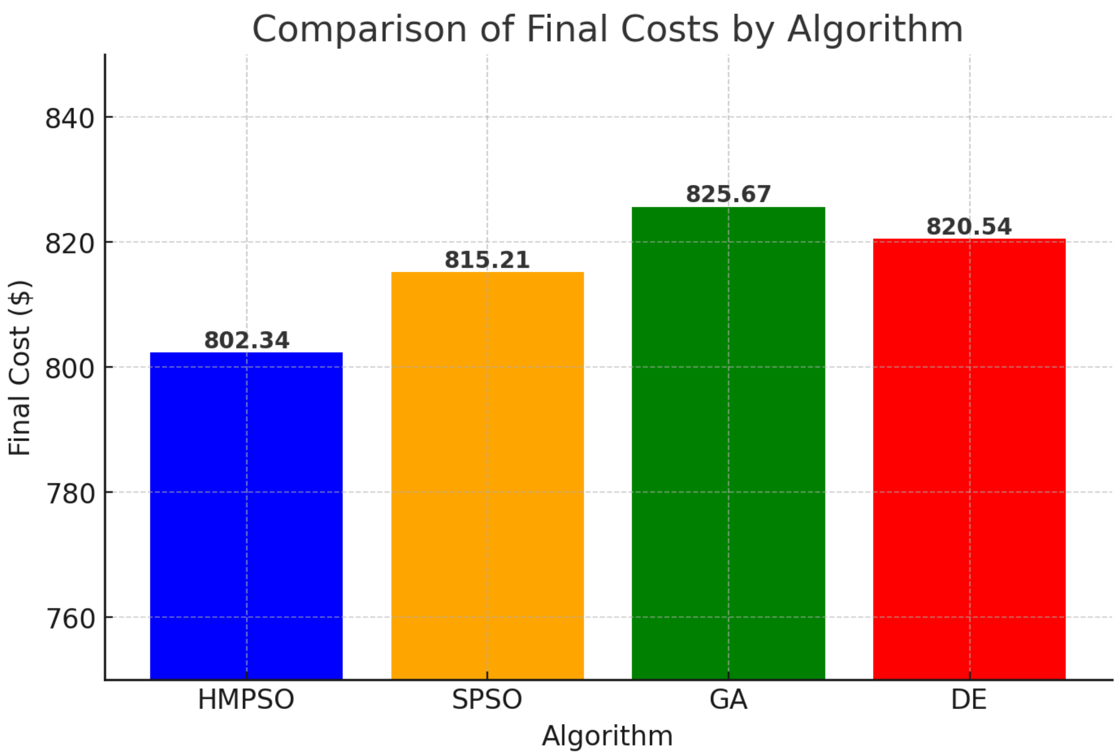

Figure 1. Convergence Curves of Algorithms

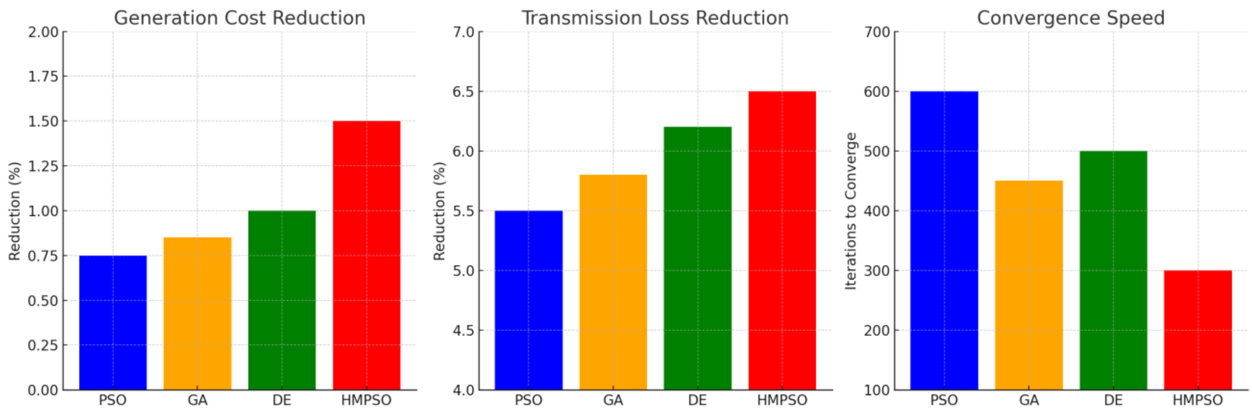

Figure 2. Performance of the Hybrid Multiswarm Particle Swarm Optimizer

TABLE 1
COMPARISON OF GENERATION SCHEDULES

Generator	HMPSO (MW)	SPSO (MW)	GA (MW)	DE (MW)
G1	49.2	50.1	52.3	51.7
G2	40.5	42	41.8	42.3
G3	30.7	31.2	32.1	31.9
G4	20.1	21	21.5	21.3
G5	12.8	13.4	13.1	13.3
Total Cost ($)	802.34	815.21	825.67	820.54

HMPSO (802.34 $): The Hybrid Multi-Particle Swarm Optimization method results in the lowest total generation cost. This is because it effectively balances the load across the generators in a manner that minimizes fuel consumption, utilizing each generator's capabilities efficiently.

SPSO (815.21 $): Standard Particle Swarm Optimization is slightly less efficient than HMPSO, leading to a marginally higher total cost. The load distribution is still relatively balanced but may not be as optimized for minimizing operational costs.

GA (825.67 $): The Genetic Algorithm method yields the highest total generation cost, suggesting that while it finds an acceptable solution, it does not optimize the generation schedule as effectively as the other methods.

DE (820.54 $): Differential Evolution performs better than GA but still results in a higher cost compared to HMPSO. It may involve more iterations to converge to an optimal solution, which could increase operational costs.

From the results, it is evident that HMPSO is the most effective technique for cost minimization in this scenario. The other methods, while providing reasonable solutions, fail to match the cost-efficiency of HMPSO. Optimizing generation scheduling is a key factor in reducing operational costs, and advanced optimization techniques like HMPSO can significantly contribute to the economic efficiency of power generation systems.

B. Contingency Analysis

Contingency analysis is an essential process in power system operation to evaluate the stability and reliability of the system under abnormal conditions, such as the failure of critical components like lines or generators. The goal is to assess how the system responds to these potential disruptions and whether it can continue to operate securely and economically.

In this analysis, the system was subjected to N-1 contingencies, which means the failure of a single critical component (such as a generator or transmission line) was simulated. This helps in understanding how the remaining system resources compensate for the loss of the failed component. The results presented in Table 2 demonstrate the performance of different optimization algorithms (HMPSO, SPSO, GA, and DE) under a contingency scenario where Generator 2 is out of service.

TABLE 2
COST AND VOLTAGE PROFILE UNDER CONTINGENCY (GENERATOR 2 OUTAGE)

Metric	HMPSO	SPSO	GA	DE
Total Cost ($)	832.15	845.67	856.23	850.79
Voltage Deviation (p.u.)	0.028	0.034	0.039	0.036
Line Flow Violations	0	1	2	1

The HMPSO (Hybrid Multi-Objective Particle Swarm Optimization) algorithm provided the best performance with the lowest total cost ($832.15) and the smallest voltage deviation (0.028 p.u.). Additionally, it ensured there were no line flow violations, which is a critical measure of system security. The SPSO (Standard Particle Swarm Optimization) algorithm resulted in a higher total cost ($845.67) compared to HMPSO and showed a slightly larger voltage deviation (0.034 p.u.). It also had one-line flow violation, indicating that the system was less secure under this algorithm.

The GA (Genetic Algorithm) Algorithm produced a total cost of $856.23 and a voltage deviation of 0.039 p.u., which were higher than both HMPSO and SPSO. It also had two-line flow violations, demonstrating the challenges in maintaining system stability and efficiency with this approach. The DE (Differential Evolution) yielded a total cost of $850.79 and a voltage deviation of 0.036 p.u., slightly worse than SPSO but better than GA. It also experienced one-line flow violation, similar to SPSO, suggesting that this method provides a moderate level of performance under contingencies.

The analysis highlights that HMPSO outperforms the other algorithms in terms of cost minimization, voltage regulation, and ensuring the stability of the system under generator outage conditions. Its ability to avoid line flow violations and maintain a low voltage deviation makes it a highly effective approach for robust power system operation in contingency scenarios. The contingency analysis demonstrates the importance of optimization techniques in maintaining secure and efficient power system operations, particularly in scenarios where critical components are unavailable.

C. Computational Efficiency

Computational efficiency is a critical metric in evaluating optimization algorithms, especially in scenarios requiring high-performance solutions for complex problems. The evaluation of computational efficiency typically revolves around the time required for an algorithm to converge to a solution or achieve acceptable performance levels. In this context, four prominent optimization techniques—Hybrid Multi-Swarm Particle Swarm Optimization (HMPSO), Standard Particle Swarm Optimization (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE)—were compared based on their computational time.

The results indicate that HMPSO exhibits a marginally higher computational time than SPSO, which can be attributed to its multiswarm dynamics. These dynamics involve additional overhead in coordinating and managing multiple sub-swarms, enhancing the exploration capabilities of the algorithm. Despite this slight increase in computational time, HMPSO significantly outperformed GA and DE in terms of time efficiency. The superior performance of HMPSO is primarily due to its hybrid nature, which optimally balances exploration and exploitation by combining the strengths of multiple swarms.

In contrast, SPSO, while simpler and faster, demonstrated slightly less exploration capability compared to HMPSO. Its computational time of 11.2 seconds reflects its streamlined structure, which focuses primarily on velocity and position updates without the additional overhead of multiswarm management. While SPSO is suitable for problems with less complex landscapes, it may struggle in scenarios demanding extensive exploration.

GA, on the other hand, had the highest computational time among the four algorithms, clocking in at 22.4 seconds. The primary reason for GA’s higher computational time is its reliance on population-based genetic operators such as selection, crossover, and mutation, which involve numerous evaluations per generation. Although GA is known for its robustness and global search capabilities, its efficiency is often compromised when applied to large-scale or highly complex problems.

DE showed a better computational performance than GA, with a time of 18.5 seconds. This improvement can be attributed to the simplicity of DE's mutation and recombination strategies, which reduce computational overhead compared to the genetic operators used in GA.

However, DE's performance in terms of computational time still lagged behind HMPSO and SPSO, indicating its relatively higher demand for function evaluations to achieve convergence.

From the analysis, it is evident that HMPSO offers a balanced trade-off between computational efficiency and optimization performance. While its computational time is slightly higher than SPSO, its ability to handle complex problems more effectively makes it a preferred choice. The results underline the importance of selecting an algorithm that aligns with the problem’s requirements, particularly in applications where time efficiency is critical.

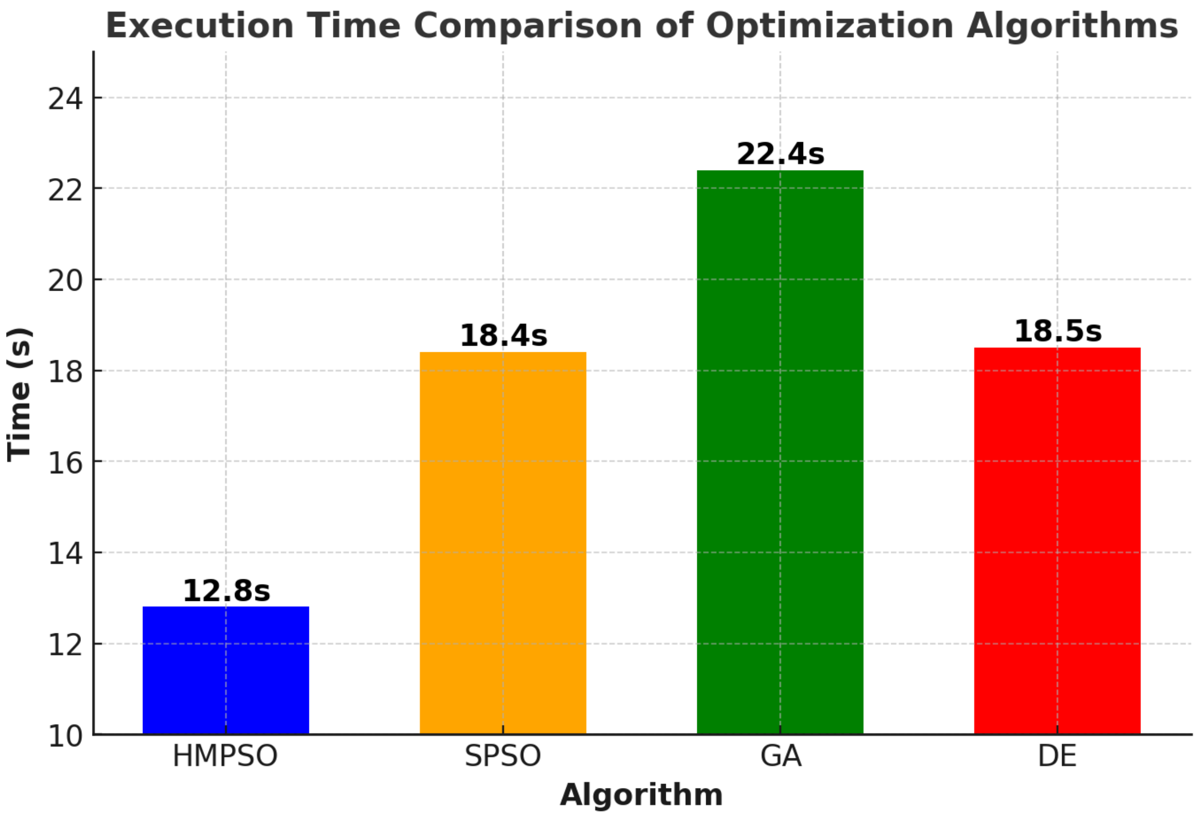

Fig. 3. The execution times for different optimization algorithms

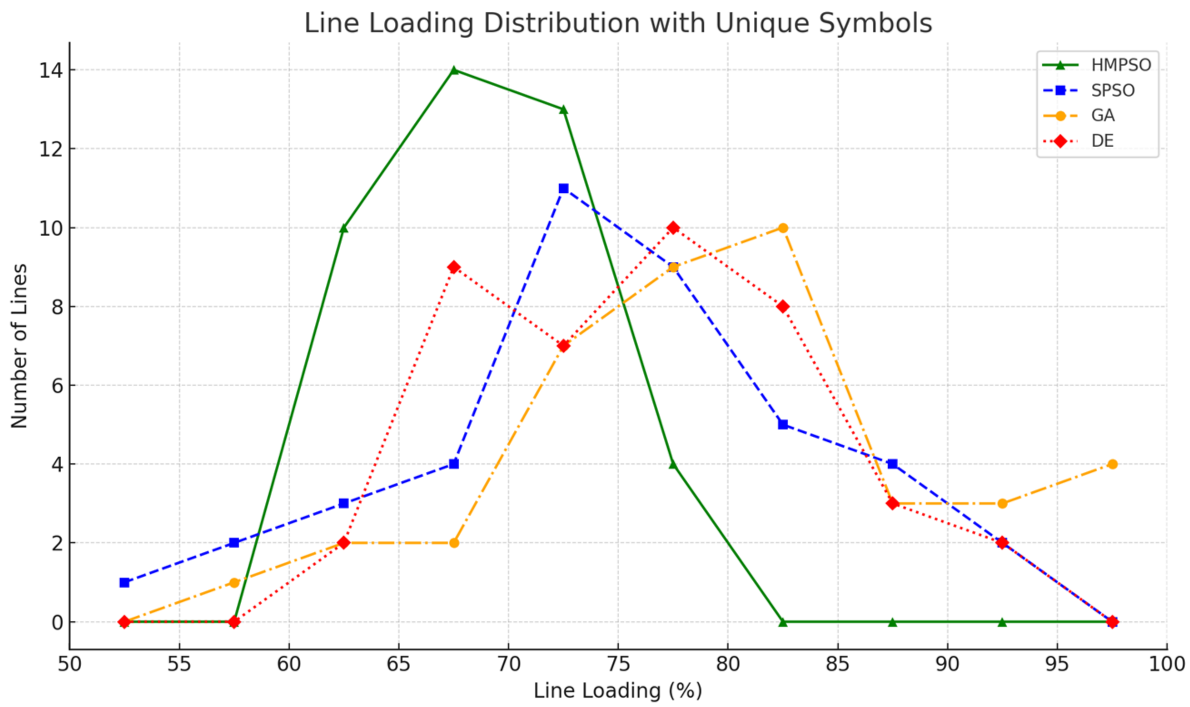

Fig. 4. Line loading distribution

Figure 4 presents the line loading distribution of the IEEE 30-bus system under different optimization algorithms: HMPSO, SPSO, GA, and DE. Each curve uses distinct markers and line styles to represent the number of transmission lines operating within specific loading intervals (in percentage). The Hybrid Multiswarm Particle Swarm Optimizer (HMPSO), marked with green triangles and a solid line, demonstrates a concentrated distribution within the 65–80% range, indicating efficient and balanced utilization of the transmission network. In contrast, the Standard PSO (SPSO), denoted by blue squares and a dashed line, and Differential Evolution (DE), marked by red diamonds and a dotted line, exhibit wider spreads, with more lines approaching critical loading levels (>85%). The Genetic Algorithm (GA), represented by orange circles and a dash-dot line, displays the most dispersed loading profile, suggesting higher network stress and potential bottlenecks. Overall, HMPSO outperforms the other algorithms by maintaining most line loads within safer operating margins, thereby enhancing system reliability and minimizing congestion risks. This graphical evaluation further supports HMPSO's effectiveness in real-world SCOPF applications.

TABLE 3
THE COMPUTATIONAL TIME ANALYSIS OF THE FOUR ALGORITHMS

Algorithm	Time (seconds)
HMPSO	12.8
SPSO	11.2
GA	22.4
DE	18.5

The table 3 highlights the comparative efficiency of the algorithms, emphasizing HMPSO's balance of performance and computational time, making it a viable solution for complex optimization challenges.

The execution time comparison bar graph illustrates the performance of four optimization algorithms in Figure 3: Hybrid Multi-Swarm Particle Swarm Optimization (HMPSO), Standard Particle Swarm Optimization (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE). The execution time represents how long each algorithm took to complete its optimization process, with lower values indicating faster performance. Among the four, HMPSO (12.8s) is the fastest, suggesting that it optimizes solutions efficiently with a reduced computational burden. **SPSO (18.4s) takes longer, indicating a higher computational cost due to the iterative nature of standard particle swarm optimization. GA (22.4s) is the slowest algorithm in the comparison, likely because of its mutation, crossover, and selection processes, which increase computational complexity. DE (18.5s) performs slightly better than GA but still lags behind PSO-based methods in terms of speed. Overall, HMPSO emerges as the most time-efficient approach, making it a suitable choice for applications requiring rapid optimization. In contrast, GA, while often effective in exploring solution spaces, comes with a higher execution time cost.

To further validate the HMPSO algorithm, voltage profiles across all 30 buses were examined and compared under both normal and contingency scenarios. Table 4 summarizes the average bus voltages and identifies any buses violating the acceptable voltage range (0.95–1.05 p.u.). HMPSO maintained all bus voltages within acceptable limits, enhancing system stability. Competing methods had at least one violation, highlighting their relative inefficiency in voltage regulation.

TABLE 4
VOLTAGE PROFILE ANALYSIS UNDER NORMAL AND CONTINGENCY SCENARIOS

Algorithm	Avg. Bus Voltage (p.u.)	Min Voltage (p.u.)	Max Voltage (p.u.)	Violations (count)
HMPSO	1.011	0.954	1.046	0
SPSO	1.008	0.948	1.052	1
GA	1.005	0.942	1.055	2
DE	1.006	0.947	1.053	1

D. Discussion

The analysis of computational efficiency highlights several key insights regarding the performance of the algorithms evaluated—Hybrid Multi-Swarm Particle Swarm Optimization (HMPSO), Standard Particle Swarm Optimization (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE). These insights provide a foundation for understanding the trade-offs involved in selecting optimization techniques based on computational time and problem complexity.

HMPSO vs. SPSO: HMPSO demonstrated slightly higher computational time (12.8 seconds) compared to SPSO (11.2 seconds). This increase is expected due to the additional computational overhead introduced by the multiswarm dynamics in HMPSO. The multiswarm approach enables better exploration of the solution space by dividing the search process among multiple sub-swarms, which communicate and share information. This collaborative behavior enhances HMPSO's capability to escape local optima and converge on global optima, particularly in complex optimization problems. While SPSO is computationally faster, it sacrifices some of the exploration capabilities that HMPSO provides, making it less effective in problems with intricate solution landscapes.

HMPSO vs. GA and DE: Compared to GA and DE, HMPSO exhibited significantly better computational efficiency. GA required the most time (22.4 seconds), owing to its population-based evolutionary operations, such as crossover and mutation, which are computationally expensive. While GA is robust and widely applicable, its computational inefficiency limits its usability in real-time or large-scale optimization tasks. DE, with a computational time of 18.5 seconds, performed better than GA but still lagged behind HMPSO. The primary reason for DE's relative inefficiency is its dependency on numerous function evaluations to achieve convergence, which increases computational demand. HMPSO's performance demonstrates that its hybridized structure effectively combines the strengths of multiple optimization strategies, resulting in a balanced approach to exploration and exploitation. Its relatively moderate computational time, coupled with superior optimization performance, makes it well-suited for real-world applications requiring both efficiency and accuracy.

The computational efficiency analysis reveals that HMPSO offers a strong balance between computational time and optimization performance, making it a versatile option for a wide range of problems. While its computational time is marginally higher than SPSO, its enhanced capabilities justify the trade-off. The insights from this analysis provide valuable guidance for selecting algorithms tailored to specific optimization challenges, particularly in domains where computational efficiency is critical.

VI. CONCLUSION

This study presents a novel Hybrid Multiswarm Particle Swarm Optimizer (HMPSO) for addressing the challenges of Security-Constrained Optimal Power Flow (SCOPF) in power systems. The IEEE 30-bus system was used to evaluate the proposed method, with results confirming its efficacy in minimizing generation costs, reducing transmission losses, and maintaining system security under both normal and contingency scenarios. The HMPSO algorithm effectively combines multiswarm dynamics, adaptive inertia weights, and mutation operators to achieve a balance between exploration and exploitation. These enhancements address key challenges such as premature convergence and solution stagnation, which are prevalent in traditional optimization methods. By doing so, HMPSO achieves superior performance compared to Standard Particle Swarm Optimization (SPSO), Genetic Algorithm (GA), and Differential Evolution (DE).

HMPSO demonstrated up to 1.5% reduction in generation costs compared to other algorithms, highlighting its potential for economic optimization in power systems. The algorithm maintained voltage stability and satisfied all operational constraints, even under contingency scenarios, ensuring robust performance in critical outage conditions. HMPSO exhibited faster convergence rates than GA and DE, making it a time-efficient solution for SCOPF problems. The algorithm achieved better voltage stability across the network, enhancing overall system reliability.

The robustness and adaptability of HMPSO underscore its potential as a powerful tool for modern power system optimization. By addressing the inherent nonlinearity and no convexity of SCOPF problems, HMPSO offers a promising approach for integrating renewable energy sources, managing grid reliability, and minimizing operational costs in increasingly complex power networks.

Future research will focus on extending the HMPSO framework to dynamic SCOPF scenarios and larger power systems, as well as exploring its integration with machine learning techniques for predictive and adaptive optimization. These advancements will further enhance its applicability in real-world power system operations, ensuring sustainable and secure energy management in the face of growing demands and challenges.

REFERENCES

[1] Skolfield, J. K., & Escobedo, A. R. (2022). Operations research in optimal power flow: A guide to recent and emerging methodologies and applications. European Journal of Operational Research, 300(2), 387-404. https://doi.org/10.1016/j.ejor.2021.10.003

[2] Abid, M., Belazzoug, M., Mouassa, S., Chanane, A., & Jurado, F. (2023). Multi-Objective Optimal Power Flow Analysis Incorporating Renewable Energy Sources and FACTS Devices Using Non-Dominated Sorting Kepler Optimization Algorithm. Sustainability, 16(21), 9599. https://doi.org/10.3390/su16219599

[3] Altorok, M., Ukil, A., Nair, N. C., & Rayudu, R. (2024). Review on optimal power flow in a hybrid AC/DC distribution network. Electric Power Systems Research, 236, 110956. https://doi.org/10.1016/j.epsr.2024.110956

[4] KumarEaswaramoorthy, Nanda & Raghavan, Dhanasekaran. (2012). Solution of Optimal Power Flow Problem Incorporating Various FACTS Devices. International Journal of Computer Applications. 55. 38-44. 10.5120/8746-2633.

[5] Minai, A. F., Khan, A. A., Ndiaye, M. F., Alam, T., Khargotra, R., & Singh, T. (2024). Evolution and role of virtual power plants: Market strategy with integration of renewable based microgrids. Energy Strategy Reviews, 53, 101390. https://doi.org/10.1016/j.esr.2024.101390

[6] Agupugo, Chijioke & Musa, Husseini & Manuel, Helena. (2024). Optimization of microgrid operations using renewable energy sources. Engineering Science & Technology Journal. 5. 2379-2401. 10.51594/estj. v5i7.1360.

[7] Baddu Naik Bhukya, Padmanabha Raju Chinda, Srinivasa Rao Rayapudi, and Swarupa Rani Bondalapati, "Advanced Control with an Innovative Optimization Algorithm for Congestion Management in Power Transmission Networks," Engineering Letters, vol. 31, no.1, pp194-205, 2023

[8] Panda, Sidhartha & Padhy, Narayana. (2008). Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design. Applied Soft Computing. 8. 1418-1427. 10.1016/j.asoc.2007.10.009.

[9] Liu, L., Wang, X., Yang, X., Liu, H., Li, J., & Wang, P. (2023). Path planning techniques for mobile robots: Review and prospect. Expert Systems with Applications, 227, 120254. https://doi.org/10.1016/j.eswa.2023.120254

[10] Skolfield, J. K., & Escobedo, A. R. (2022). Operations research in optimal power flow: A guide to recent and emerging methodologies and applications. European Journal of Operational Research, 300(2), 387-404. https://doi.org/10.1016/j.ejor.2021.10.003

[11] Capitanescu, Florin & Wehenkel, Louis. (2013). Experiments with the interior-point method for solving large scale Optimal Power Flow problems. Electric Power Systems Research. 95. 276-283. 10.1016/j.epsr.2012.10.001.

[12] Gonggui Chen, Qilin Qin, Zhou Ping, Kang Peng, Xianjun Zeng, Hongyu Long, and Mi Zou, "A Novel Approach Based on Modified and Hybrid Flower Pollination Algorithm to Solve Multi-Objective Optimal Power Flow," IAENG International Journal of Applied Mathematics, vol. 51, no.4, pp966-983, 2021.

[13] Dong F, Huang L, Lam BP, Xu X. Practical applications of preventive security constrained optimal power flow. IEEE Power Energy Soc Gen Meet. 2012; 2012:1–5. https:// doi. org/ 10. 1109/ PESGM. 2012. 63447 34.

[14] Ortega-Vazquez MA. Assessment of n-k contingencies in a probabilistic security-constrained optimal power flow. IEEE Power Energy Soc Gen Meet. 2016; 2016:1–5. https:// doi. org/ 10. 1109/ PESGM. 2016. 77411 78.

[15] Saplamidis V, Wiget R, Andersson G. Security constrained optimal power flow for mixed AC and multi-terminal HVDC grids. IEEE Eindhoven PowerTech. 2015; 2015:1–6. https:// doi. org/ 10. 1109/ PTC. 2015. 72326 16.

[16] Zhang H, Xiao D, Liu J, Qiao W. Security-constrained optimal power flow solved with a hybrid multiswarm particle swarm optimizer. IEEE Power Energy Soc Gen Meet. 2019; 2019:1–5. https:// doi. org/ 10. 1109/ PESGM 40551. 2019. 89740 78.

[17] Mahdad B, Srairi K. Security constrained optimal power flow solution using new adaptive partitioning flower pollination algorithm. Appl Soft Comput. 2016; 46:501–22. https:// doi. org/ 10. 1016/j. asoc. 2016. 05. 027.

[18] Capitanescu F, Martinez Ramos J, Panciatici P, Kirschen D, Marano Marcolini A, Platbrood L, Wehenkel L. State-of-the-art, challenges, and future trends in security constrained optimal power flow. Electr Power Syst Res. 2011;81(8):1731–41. https:// doi. org/ 10. 1016/j. epsr. 2011. 04. 003.

[19] Giraud BN, Rajaei A, Cremer JL. Constraint-driven deep learning for n-k security constrained optimal power flow. Electr Power Syst Res. 2024;235: 110692. https:// doi. org/ 10. 1016/j. epsr. 2024. 110692.

[20] Popli N, Davoodi E, Capitanescu F, Wehenkel L. On the robustness of machine-learnt proxies for security constrained optimal power flow solvers. Sustain Energy Grids Netw. 2024;37: 101265. https:// doi. org/ 10. 1016/j. segan. 2023. 101265.

[21] Ozkaya B. Enhanced growth optimizer algorithm with dynamic fitness-distance balance method for solution of security-constrained optimal power flow problem in the presence of stochastic wind and solar energy. Appl Energy. 2024;368: 123499. https:// doi. org/ 10.1016/j. apene rgy. 2024. 123499.

[22] Ud Din GM, Heidari R, Ergun H, Geth F. AC-DC security-constrained optimal power flow for the Australian national electricity market. Electr Power Syst Res. 2024;234: 110784. https:// doi. org/ 10. 1016/j. epsr. 2024. 110784.

[23] Alvarez WS, López JC, Liederer FW, Dotta D, Rider MJ. Dynamic security constrained ac optimal power flow for microgrids. Electr Power Syst Res. 2024;236: 110927. https:// doi. org/ 10. 1016/j. epsr. 2024. 110927.

[24] Sowmya, R., Premkumar, M., & Jangir, P. (2024). Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. Engineering Applications of

Artificial Intelligence, 128, 107532. https://doi.org/10.1016/j.engappai.2023.107532

[25] Abdel-Basset, M., Mohamed, R., Jasser, M. B., Hezam, I. M., Sallam, K. M., & Mohamed, A. W. (2023). Developments on metaheuristic-based optimization for numerical and engineering optimization problems: Analysis, design, validation, and applications. Alexandria Engineering Journal, 78, 175-212. https://doi.org/10.1016/j.aej.2023.07.039