

Comparative Analysis of Optimization Techniques for Improvement of Smart Grid Performance and Reliability

Jumshed Akhtar

Faculty of Engineering, Lincoln university college, Petling Jaya.Malaysia

Moazam Shehzad

Faculty of Engineering, Lincoln university college, Petling Jaya.Malaysia

University of Engg. And Tech. Lahore. Lahore, Pakistan

Abdul Manan

University of Engg. And Tech. Lahore. Lahore, Pakistan, Pakistan

Hammad Ali Zahid

University of Engg. And Tech. Lahore. Lahore, Pakistan

ABSTRACT: The increased complexity of modern power systems, driven by the integration of renewable energy sources and growing electricity demand, has led to optimized smart grids. Traditional grid management approaches are often insufficient in addressing the challenges of efficiency, reliability, and minimizing losses. This study evaluates the performance of various optimization algorithms in minimizing power losses, reducing voltage deviation, and improving convergence efficacy in power system optimization. Among the analyzed methods, Linear Programming (LP) exhibits the highest power losses of 220 kW and voltage deviation, making it the least effective approach. In contrast, Genetic Algorithm (GA), Traditional Particle Swarm Optimization (PSO), and Differential Evolution (DE) achieve moderate reductions in power losses (10–18%) and voltage deviation (12–20%). The Proposed Deep Reinforcement Learning-based PSO (DRL-PSO) and Improved PSO demonstrate enhanced performance, reducing power losses by 22–28% and improving voltage stability. The most efficient methods—Multi-Objective PSO (MOPSO), Multi-Objective Wind Driven Optimization (MOWDO), and Multi-Objective Genetic Algorithm (MOGA)—achieve the lowest total losses (~140–160 kW) with a significant reduction of 30–35% and minimal voltage deviation (~0.01–0.015 p.u.), ensuring optimal system stability. While DRL-PSO requires longer convergence time due to its complexity, MOPSO, MOWDO, and MOGA exhibit the fastest convergence while maintaining superior optimization performance, making them suitable for real-time applications

Introduction

Literature Review

Result and Discussion

Conclusion

References

1. Introduction

Today, 80% of World's energy system is supplied by fossil fuels such as oil, coal and gas. Fossil fuels contain a large amount of carbon which causes more heat and pollution in the environment. In result, it is causing climate change and global warming. Global warming is a worldwide problem. To overcome global warming Sustainable Development Goals were introduced [1]. Sustainable development goals (SDGs) are set of 17 goals which are adopted by the United Nations in 2015 for the promotion of sustainable development, innovations and

addressing global challenges. SDG 7 refers to affordable and clean energy and SDG 9[2] refers to Industry, Innovation and Infrastructure. As the aim of these goals, we ensure public access to energy with negligible losses. So, energy consumers are required to manage demand, technical and non-technical issues and technical characteristics of grids which have become more complex from the past few years shifting from traditional centralized production to prosumer model that is connected to distribution. This evolution has been facilitated by establishing suitable infrastructure for communication that is smart grid [3]. Smart grid is a modern electrical grid that uses information and communication technologies to manage electricity demand and utilities in an efficient, reliable and sustainable way. In traditional power systems the main challenges are transmission line losses, transformer losses and technical losses. According to studies technical losses are up to 10% of the total generation while there are also non-technical challenges such as energy theft, error metering and complexity in grid management. To overcome these challenges traditional methods are used such as linear programming (LP), mixed-integer programming (MIP) [4]. There are some problems that exist in traditional techniques such as in Linear Programming, algorithm needs to be more flexible. In this way, LP can't avoid errors. In Non-Linear Programming, algorithms require Differentiable Continuous Objective Function, which limits its scope. In Dynamic Programming, algorithms could face dimensionality collapse while solving high dimensional problems [5]. Heuristic techniques such as Genetic Algorithm (GA) Particle Swarm Optimization (PSO), Simulated Annealing Algorithm (SSA), Chaos Optimization Algorithm (COA), Artificial Fish Algorithm (AFA), Whale Optimization Algorithm (WOA), and Salp Swarm Algorithm (SSA) have been applied to different problems. These methods can help me struggle with scalability, adaptability and computational overhead. Artificial Intelligence (AI) based models such as Machine Learning (ML), Deep Learning (DL) and Reinforce Learning (RL) when combined to Optimization techniques result in robust solutions that reduces complexity of smart grids, faster convergence and greater adaptability. These AI based models learn from historical data and predict future state of smart grids [4].

This paper presents a hybrid AI-based optimization framework that integrates multiple optimization techniques, including Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE), Deep Reinforcement Learning combined with PSO (DRL-PSO), Improved PSO, Multi-Objective PSO (MOPSO), Multi-Objective Wind Driven Optimization (MOWDO), and Multi-Objective Genetic Algorithm (MOGA). These algorithms collectively provide a robust mechanism for dynamically adjusting control strategies based on real-time grid conditions while ensuring global optimization of key variables such as generation set-points, voltage levels, and load distribution. The proposed method is tested on the IEEE 33-bus benchmark system, demonstrating significant improvements in loss reduction, voltage regulation, and computational efficiency compared to traditional methods.

The remainder of this paper is organized as follows: Section 2 provides a comprehensive literature review, focusing on traditional and AI-based optimization techniques for smart grids. Section 3 discusses system modeling, objective functions, and constraints. Section 4 introduces the formulation of the proposed optimization framework. Section 5 presents the case study using the IEEE 33 bus system, while Section 6 provides a comparative analysis of the results. Finally, Section 7 concludes the paper and discusses future research directions.

2. Literature Review

Ali et al. (2024) proposed a novel AI-based optimization framework for smart grids, designed to reduce transmission and distribution losses, enhance system efficiency, and improve grid reliability. Their study conducts a comparative analysis of traditional optimization techniques and state-of-the-art AI algorithms, demonstrating the superiority of the proposed method using the IEEE 33-bus benchmark system. Results indicate a significant reduction in power losses and an improvement in voltage profiles, leading to a more efficient and reliable grid. [4] Reactive power optimization is another critical aspect of smart grid optimization, as it directly impacts voltage stability and power loss reduction. The difficulties with PSO-based optimization are highlighted by D. Mourtzis and J. Angelopoulos (2023), specifically the problems of early convergence and local optima entrapment. They suggest an enhanced PSO algorithm created especially to reduce power loss in smart grids to overcome these constraints. Their findings, which were benchmarked using the IEEE 30-bus standardized model, show an 11% improvement over traditional PSO techniques, underscoring the potential of improved metaheuristic techniques for smart grid reactive power optimization. [3]

Razavi et al. (2020) present a comprehensive approach to improving the daily performance of ADNs, focusing on four key objectives: (i) reducing active losses, (ii) improving the voltage profile, (iii) enhancing network reliability, and (iv) minimizing distribution network operation costs. Their method uses real-time data at the start of each day to solve the optimization problem efficiently, considering the likelihood of renewable resource failures. They present an enhanced Crow Search Algorithm (ICSA) to get optimal performance, and its efficacy in maximizing ADN performance is confirmed on the IEEE 33-bus radial distribution system. [6]

Fu et al. (2024) suggest an adaptive reactive power optimization control technique in offshore wind farms (OWFs). Reactive power margins are improved, active power network losses are decreased, and voltage variations at wind turbine (WT) terminals are minimized using their multi-objective optimization framework. The suggested adaptive optimization technique successfully stabilizes node voltages in OWFs while increasing overall efficiency, according to simulation results. Their suggested Unified Adaptive PSO (UAPSO) method shows a promising approach for optimizing reactive power control in offshore wind applications, with an approximate 10% gain in solution time and improved accuracy when compared to the classic PSO algorithm. [7]

Z. Li and J. Xiong (2024) introduce an improved version of the multi-objective Particle Swarm Optimization (MOPSO) algorithm, specifically designed to address the unique challenges of reactive power optimization in modern distribution networks. According to the test findings, this algorithm can produce a sizable number of Pareto optimal solutions, offering a varied and evenly distributed collection of reactive compensating techniques. When used in a case study including a 33-node network, the enhanced MOPSO algorithm shows notable benefits in reactive power optimization. The optimization findings demonstrate the viability and efficacy of the suggested approach, providing distribution networks with a more consistent and varied selection of reactive compensating options. [8]. Carrascal et al. (2024) explored the extension of DEN2DE, a flexible routing and reconfiguration solution that may be useful for smart grids (SGs), using AI-based failure prediction. Real-world datasets and

randomly generated topologies based on the IEEE 123-Node Test Feeder are used in the study. In particular, Random Forest (RF) and Support Vector Machine (SVM) are evaluated as machine learning (ML) approaches, and Artificial Neural Networks (ANN) are evaluated as deep learning (DL) techniques. Recall, accuracy, and precision are used to evaluate models. With 94.28% precision and 81.05% recall, the RF model with Recursive Feature Elimination (RFECV) outperforms SVM (precision: 89.32%, recall: 6.95%) and ANN (precision: 72.17%, recall: 13.49%) in terms of defect detection accuracy and dependability, according to the results. [9]

Kadar (2013) discusses various optimization techniques applied in power system control, highlighting both traditional methods (like linear programming) and AI-based approaches (such as genetic algorithms and particle swarm optimization). The paper addresses key challenges in modern energy systems—ranging from renewable integration to market deregulation—and presents solutions for tasks like energy storage, network reliability, power mix optimization, and regional energy trade through multi-objective and rule-based models [10]. Gundi and Rajarajeswari (2014) present a cost optimization method for smart grids using a Genetic Algorithm (GA), focusing on minimizing short-term time-averaged electricity costs for inelastic loads. The system integrates renewable energy, battery storage, and real-time pricing, with GA optimizing variables such as battery discharge, grid energy usage, and renewable energy allocation. Simulation results show that increasing battery capacity significantly reduces electricity costs, demonstrating the efficiency of GA in smart grid cost management [11].

Kumar and Naik (2017) propose a 24-hour load shifting strategy for smart grids using Particle Swarm Optimization (PSO) to minimize peak demand and electricity costs. The method reallocates shift able residential, commercial, and industrial loads from peak to off-peak hours based on real-time electricity prices. Compared to Genetic Algorithms (GA), PSO achieved greater reductions in peak-to-average load ratios and operational costs. The simulation showed PSO outperforms GA in all three sectors, making the smart grid more efficient, reliable, and cost-effective [12]. The paper reviews the concept, framework, benefits, challenges, and optimization strategies of Vehicle-to-Grid (V2G) technology. The increasing adoption of electric vehicles (EVs) as an alternative to traditional internal combustion engine vehicles is driven by energy crises and environmental concerns. The development of smart grid technology has enabled EVs to play a role in improving power system operations through V2G technology. V2G allows bidirectional energy exchange between EVs and the power grid, providing services such as power grid regulation, spinning reserve, peak load shaving, load leveling, and reactive power compensation [13].

This paper addresses the issue of the underutilization of generation capacity in power grids, which is mainly due to the difference between peak and off-peak load demand. The authors propose using demand response (DR) in smart grids to allow users to adjust their energy consumption based on costs, aiming to reduce peak load demand and increase off-peak demand. They formulate the load scheduling problem as a constrained multi-objective optimization problem (CMOP) with the objectives of minimizing energy consumption cost and maximizing utility. The paper then develops two evolutionary algorithms (EAs) to find Pareto-front solutions for this problem [14]. This paper introduces a novel energy optimization model

designed to enhance the performance of smart micro grids that incorporate renewable energy sources (RES) like solar and wind power. The model addresses the challenges of energy optimization, such as the unpredictability of RES and consumer behaviour, by using a hybrid scheme of demand response programs (DRPS) and incline block tariff (IBT) and by using multi-objective optimization algorithms. The model aims to reduce operating costs and pollution emissions while maximizing the use of RES [15].

This paper proposes a hybrid genetic algorithm (GA)-particle swarm optimization (PSO) algorithm to solve a multi-objective demand side management (MODSM) problem in a smart grid that includes wind power. The algorithm aims to minimize electricity costs and optimally allocate generation and load demand in a day-ahead market. The hybrid algorithm combines the strengths of GA and PSO to improve convergence speed and avoid local minimum, using a fusion factor to balance exploration and exploitation [16].

This paper reviews various stochastic programming (SP) techniques used by researchers to address uncertainties in renewable energy sources and load demands within smart grids. It discusses the formulations of different researchers for objective functions like cost, loss, generation expansion, and voltage control. The paper also explains SP variants such as the recourse model, chance constrained programming (CCP), sample average approximation (SAA), and risk aversion, including their applications and mathematical expressions [17]. This paper addresses the problem of optimizing electricity costs within a smart grid environment using Demand Side Management (DSM). It formulates a multi-objective problem that considers the conflicting interests of the utility, the demand response aggregator, and the customers in a day-ahead market. The Particle Swarm Optimization (PSO) algorithm is employed to optimize the generation and load patterns, aiming to minimize electricity costs. The effectiveness of the proposed approach is demonstrated by comparing the results with an existing optimization technique [18].

This paper outlines new approaches in grid protection and optimization of distribution grid operation within smart grids. It discusses optimization to reduce grid losses by setting an optimal power factor at distributed energy resources (DER) and the challenges of distributed generation on grid protection [19]. This paper discusses voltage control, grid reconfiguration, and adaptive protection as key control applications in smart grids that heavily utilize information and communication technology. These control applications often involve multiple, sometimes conflicting objectives, necessitating the use of multi-objective optimization techniques. The presence of non-convex objectives and constraints favours the use of evolutionary algorithms for solving these control problems. The paper argues that genetic algorithms are well-suited for achieving control objectives in smart grids, especially with the integration of distributed generators, which introduces additional factors to consider in the optimization process [20]. This paper introduces a novel approach to enhance efficiency and reliability in smart grids using a combination of Deep Reinforcement Learning (DRL) and Particle Swarm Optimization (PSO). The research focuses on optimizing smart grid operations; specifically addressing power losses and grid reliability, and its effectiveness is demonstrated using the IEEE 33-bus system [4].

This paper reviews the application of heuristic optimization methods to enhance energy efficiency within power systems. The need for improved energy efficiency in the European Union, driven by the 2009/28/EC Directive, is the paper's foundation. The complexity of power systems often makes exact optimization techniques impractical, thus making heuristic methods a valuable tool for finding effective solutions. The paper explores how these methods are applied in transmission and distribution systems to achieve better energy efficiency [21].

This study evaluates optimization algorithms for modern power systems, tackling challenges from renewable energy integration and growing demand. It assesses algorithms for minimizing power losses, reducing voltage deviation, and improving convergence efficacy in smart grids, addressing traditional grid management shortcomings.

2.1 System Modelling:

2.1.1 Electrical Unit Integration in Smart Grids:

Concepts from a standard smart grid were developed by the National Institute for Standards and Technology (NIST) and include "architecture, architecture process, energy services interface, functional requirement, harmonization, interchangeability, and interoperability." [22]. The optimal integration problem in distribution, transmission, and micro grid networks is connected to all these ideas. Their involvement is contingent upon (i) the quantity of integrated units, (ii) the kind of units, and (iii) the style of operation. To provide demand side management in a smart grid, RES-based DG units, for instance, can be integrated to only service the grid during peak hours or combined with Energy Storage Systems (ESS) to store energy for periods of high demand. By using frequency fluctuation, the ESS unit can be utilized to smooth out power output [23] [24]. Figure 1 illustrates how EVs, solar photovoltaic (PV) plants, wind turbine plants, and BESS are integrated into a distribution network [25][26].

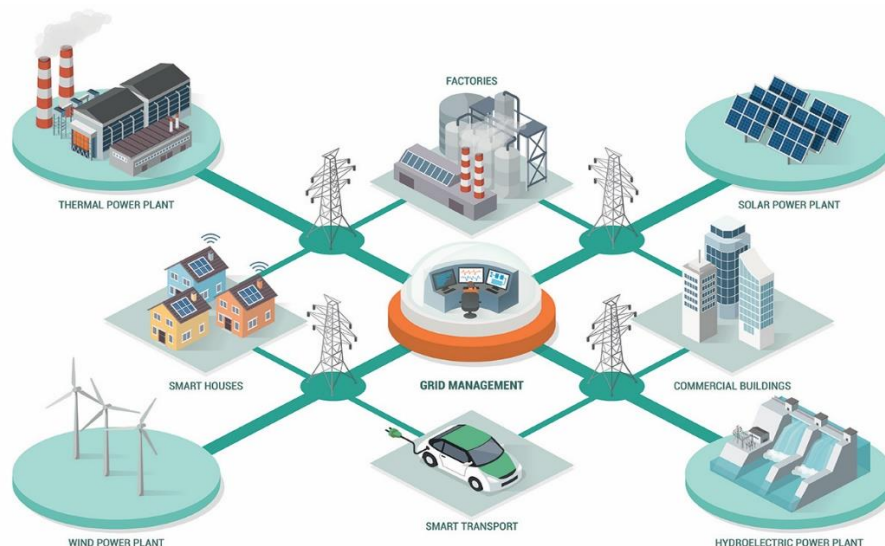
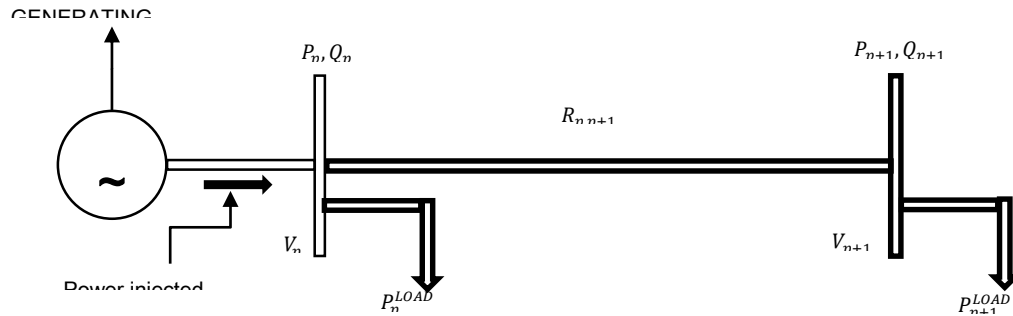


Fig 2:1 Configuration of a distribution network with the integration of different electrical units

2.1.2 Common Objectives Functions: Power Loss Minimization

The primary mathematical formulas used to model a system's properties are called objective functions. Real power loss minimization is the most frequently used goal function in optimum integration problems. Other objective functions include improving voltage stability and profile, minimizing costs, and maximizing profits.

An important component of distribution systems is power flow. Optimal power flow (OPF), continuous power flow (CPF), probabilistic power flow (PPF), and other approaches are examples of power flow techniques. To calculate power losses and voltage stability on power bus lines, several techniques have been employed to examine the components of the line. Equations for power loss, however, will depend on the kind of generator source. For example, the output power of diesel generators will defer from the power from PVs or WT. Also, a single- or three-phase type will change the model of its power flow. The schematic single line of power flow model is shown in Figure 3:2 [27][4].



2:2 schematic single line of power flow model

$$P_{p+1} = P_p - P_{p+1}^{LOAD} - R_{p,p+1} \frac{(P_{p,p+1}^2 - Q_{p,p+1}^2)}{|V_p|^2} \quad (1)$$

$$Q_{p+1} = P_p - Q_{p+1}^{LOAD} - X_{p,p+1} \frac{(P_{p,p+1}^2 - Q_{p,p+1}^2)}{|V_p|^2} \quad (2)$$

$$|V_{p+1}|^2 = |V_p|^2 - 2(R_{p,p+1}^2 P_{p,p+1}^2 + X_{p,p+1}^2 Q_{p,p+1}^2 + (R_{p,p+1}^2 + X_{p,p+1}^2) \frac{(P_{p,p+1}^2 - Q_{p,p+1}^2)}{|V_p|^2}) \quad (3)$$

Where p is the sending bus and p + 1 is the receiving bus. P, Q, and V are respectively the real power, reactive power, and voltage at bus p or p+1, while R and X are the resistance and reactance at branch p, p + 1.

The total power loss of the system is represented in (4)

$$P_{\text{total}}^{\text{loss}} = \sum_{p=1}^{n-1} P_{p,p+1}^{\text{loss}} \quad (4)$$

Voltage Stability Improvement

An essential component of loaded distribution networks is voltage stability. A voltage collapse may result from an abrupt rise or drop in the network. Voltage instability will also result from the inability to compensate for reactive power loss [28], particularly in micro grids with limited networks. An indication of network stability and load capacity, the Voltage Stability Index (VSI) can be utilized alone to solve distribution system planning problems. For distribution network design, VSI has been used in conjunction with meta-heuristic techniques throughout time. In 5, a typical equation for VSI [3] is provided.

$$VSI_{(p+1)} = V_p^4 - 4[P_{p+1}X_p - Q_{(p+1)}R_p]^2 - 4[P_{p+1}R_p + Q_{(p+1)}X_p]^2 V_p^2 \quad (5)$$

2.1.3 Constraints

The constraints used to solve this problem are the following: [7][8][5][6]:

Power Flow Equations: The real and reactive power flows in the network are governed by the following nonlinear equations:

$$P_i - P_{Di} = V_i \sum_{j=1}^n V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (6)$$

$$Q_i - Q_{Di} = V_i \sum_{j=1}^n V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad (7)$$

where: - P_i , Q_i are the real and reactive power injections at bus i , - P_{Di} , Q_{Di} are the real and reactive power demands at bus i , - G_{ij} , B_{ij} are the conductance and susceptance between buses i and j , - θ_{ij} is the voltage angle difference between buses i and j .

Voltage Limits: Each bus voltage must remain within the specified bounds:

$$V_{\min} \leq V_i \leq V_{\max} \quad \forall i \quad (8)$$

Generated Power Constraints: The generated power output must also be constrained within its limits:

$$P_{G\min} \leq P_{Gi} \leq P_{G\max} \quad \forall i \quad (9)$$

where G is the set of all generator buses.

2.2 Problem Formulation:

2.2.1 Genetic Algorithm (GA)

For systematic optimization, GA is a biological evolutionary heuristic search technique [29]. When restrictions are explicit and variables are discrete, GA might be helpful. A population of strings that match the system criteria will be generated at random by GA. Initial GA parameters, including population size, crossover probability, mutation probability, and generation size, must be established. The search will be conducted by GA using those seed strings. In this work, the distribution of loads throughout phases is represented by a chromosomal string, whereas the loads in phases A, B, and C are represented by genes. The gene is encoded as 0, 1, and 2 to reflect loads in phases A, B, and C in order to limit the search space. Subject to the limitations, each string will result in the minimization of the fitness function and indicates system status. When the fitness function value becomes negative or the load flow diverges, this string will be removed. Table 4:1 provides a summary of the Genetic Algorithm GA [30].

Table 2:1 provide a summary of GA algorithms

Algorithm I: Genetic Algorithm
Step 1: Set initial value of population and other parameters.
Step 2: Generate population randomly.
Step 3: Evaluate population with all candidate solutions.
Step 4: Generate the lbest ψ^* with its objective fitness.
Step 5: While (the end criterion is not satisfied),
SELECT parents;
RECOMBINE pairs of parents
MUTATE the resulting children;
EVALUATE children;
SELECT individuals for the new generation If $\psi > \psi^*$ $\psi^* = \psi$
end if
End while
Step 6: Output best solution ψ^*

2.2.2 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a stochastic population-based algorithm which was originally introduced by Kennedy and Eberhart [31]. A swarm of particles, each representing a possible solution, travels over the search area in PSO. Each particle's position is updated according to its velocity, which is determined by the best position the swarm as a whole and the best position it has discovered thus far. The particle in this study is a vector made up of components that require optimization, such as loads per bus. A swarm is a particle matrix. In this study, each element is referred to as a swarm member. A random particle swarm and a random speed matrix with the same dimension as the swarm are used as the initial inputs for the algorithm. Random values between 0 and 1 are entered into the speed matrix. By meeting the limitations, each particle is assessed based on the fitness function for a predetermined number of generations. The suggested algorithm's goal is to determine the ideal loads for each phase to reduce the VUF value. The P_{Best} matrix and G_{Best} vector are

produced following the evaluation of every particle from a single generation. The matrix comprising each particle's greatest performance up until generation j (the current iteration) is called P_{Best} . Up to generation j , G_{Best} was the vector with the best particle ever found. If the best particle is no longer present in the swarm, it is reintroduced. Until the maximum number of generations is reached, the algorithm will calculate the particle's new position and continue the procedure. Table 4:2 provides a summary of particle swarm optimization (PSO) [32].

Table 2:2 provide the summary of PSO algorithm

Algorithm II : Particle Swarm Optimization Algorithm
Step I: Initialize the value of acceleration constants c_1 , c_2 and swarm size.
Step II: set the counter $\alpha = 0$.
Step III: Generate random x_γ^α and $v_\gamma^\alpha \in [L, U]$ where $\gamma = 1 \dots SS$.
Step IV: Evaluate the fitness function $f(x_\gamma^\alpha)$.
Step V: Set $gbest^\alpha$. [$gbest^\alpha$ Is the best local solution in the swarm].
Step VI: Set $pbest_\gamma^\alpha$. [$pbest_\gamma^\alpha$ Is the best local solution in the swarm].
Step VII: Repeat.
Step VIII: $V_\gamma^{(\alpha+1)} = V_\gamma^{(\alpha)} + C_1 \times rand_1 \times (pbest_\gamma^\alpha - x_\gamma^\alpha) + C_2 \times rand_2 \times (gbest^\alpha - x_\gamma^\alpha)$. [$rand_1$ and $rand_2$ are random vectors $\in \{0,1\}$].
Step XI: $x_\gamma^{(\alpha+1)} = x_\gamma + V_\gamma^{(\alpha+1)}$, $\gamma = 1, \dots, SS$. [Update particles positions].
Step X: Evaluate the fitness function $f(x_\gamma^{(\alpha+1)})$, $\gamma = 1, \dots, SS$.
Step XI: if $f(x_\gamma^{(\alpha+1)}) \leq f(pbest_\gamma^{(\alpha)})$ then $pbest_\gamma^{(\alpha+1)} = x_\gamma^{(\alpha+1)}$ Else $pbest_\gamma^{(\alpha+1)} = pbest_\gamma^{(\alpha)}$ End if if $x_\gamma^{(\alpha+1)} \leq (gbest^{(\alpha)})$ then $gbest^{(\alpha+1)} = x_\gamma^{(\alpha+1)}$ Else $gbest^{(\alpha+1)} = gbest^{(\alpha)}$ End if
Step XII: increment iteration $\alpha \alpha = +1$ until satisfy the criteria.
Step XIII: Generate best particle.

2.2.3 Differential Evolution (DE):

Stron and Price introduced the Differential Evolution Algorithm (DE) in 1997 [33]. A D-dimensional vector is displayed as the DE algorithm's solution. The population size N and D dimensional vectors that DE creates at random can be written as follows [34]:

$$x_i(t) = \{x_{i,1}(t), x_{i,2}(t), \dots, x_{i,D}(t)\}, \quad [i=1, 2, \dots, N]$$

Where t is number of generations, D is number of dimensional variable and N is population size.

Through the mutation and crossover activities, a trial vector is created. The expression for the mutant vector is:

$$v_i^{(t)} = x_{\delta_1}^{(t)} + F \cdot (x_{\delta_2} - x_{\delta_2}) + F \cdot (x_{\delta_4} - x_{\delta_5})$$

Where F is the mutation scaling factor within the range [0, 1]. The second step is to create the trail vector by performing crossover between the mutant vector and the target vector, which can be expressed as follows: δ_d , $d=1, 2, 3, \dots, 5$ represent the random and mutually different integers generated within the range [1, N] and not equal to i.

$$u_{ij} = \begin{cases} v_{i,j}, & \text{if } \text{random}(0,1) \leq CR \text{ or } j = j_{\text{random}} \\ x_{i,j} & \text{otherwise} \end{cases}$$

CR, also known as the control parameter, is the crossover probability of generating parameters for a trail vector from the mutant vector within the range [0, 1]. j_{random} is a random number within the range [1, N].

The goal of the selection stage is to preserve the vector that has a higher fitness value than the mutant and trail vectors. The selection operator is defined as follows:

$$x_i^{(t+1)} = \begin{cases} u_i^t, & \text{if } f(u_i^t) \leq f(x_i^t) \\ x_i^t & \text{otherwise} \end{cases}$$

This DE is summarized in table 4:3.

Table 2:3 DE Algorithm Summary

Algorithm III: Differential Evolution Algorithm	
Step 1: Initialize the generation counter t: =0.	
Step 2: Initialize the mutation factor F and crossover probability CR.	
Step 3: Generate initial random population p *	
Step 4: Evaluate the fitness function for all individuals in p *	
Step 5: repeat.	
Step 6: for $i = 0; i < N; i++$ do	
Select random indexes $\delta_1, \delta_2, \delta_3,$	
$v_i^{(t)} = x_{\delta_1}^{(t)} + F \cdot (x_{\delta_2} - x_{\delta_2}) + F \cdot (x_{\delta_4} - x_{\delta_5})$	
$j = \text{random}(1, D)$	
for $k = 0; k < D; k++$ do	
if $\text{random}(0,1) \leq CR$ or $k = j$	
Then	$u_{ik}^{(t)} = v_{ik}^{(t)}$
Else	
	$u_{ik}^{(t)} = x_{ik}^{(t)}$
End if	
End for	
	if $f(u_i^{(t)}) \leq f(x_i^{(t)})$ then
	$x_i^{(t+1)} = x_i^{(t)}$
End if	
End (if conditions are satisfied).	

2.2.4 Hybrid DRL-PSO Algorithm

To determine the best course of action for control given the present grid state, the hybrid DRL-PSO algorithm [4] first uses DRL. To further minimize the goal function (power loss) and guarantee global optimization, these actions are subsequently improved using PSO. The integrated framework makes use of the advantages of both approaches: PSO's capacity to optimize the solution and DRL's real-time adaptation. DRL Agent: The DRL agent learns the best power flow actions using a deep neural network. The agent engages with the smart grid environment, obtaining status data (such as power flow and voltage levels) and acting to reduce losses. High losses and voltage violations are intended to be penalized by the reward function. The state-action pair (st, at) at time step t is updated according to the Bellman equation:

$$Q_{(st,at)} = r_t + \gamma \max_a Q_{(st+1,a)}$$

Where r_t the reward is received at time and is γ the discount factor.

PSO Optimization: PSO is used to fine-tune the control variables (e.g., generator set-points, capacitor switching) by optimizing the objective function Ploss. Each particle in the swarm represents a potential solution, and particles adjust their positions based on their individual and swarm-wide best experiences:

$$v_i(t+1) = wv_i(t) + c_1r_1[p_i^{best} - x_i(t)] + c_2r_2[g^{best} - x_i(t)],$$

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Where v_i is the velocity of particle i, x_i is the position, p_i^{best} is the individual best position, and g^{best} is the global best position. Working process of hybrid DRL-PSO shown in table 4:4.

Table 2:4 Working Process Of DRL-PSO Algorithm

Algorithm IV: DRL-PSO Algorithm
<p>Step I: Environment Setup:</p> <ul style="list-style-type: none"> Define the state of the grid (bus voltages, power flows, generator outputs). Establish parameters, encompassing voltage thresholds, load requirements, and restrictions on power generation.
<p>Step II: Deep Reinforcement Learning (DRL) Process:</p> <ul style="list-style-type: none"> State Input: The DRL agent assesses the present condition of the grid. Action Selection: The DRL agent selects an action (e.g., adjusting voltage set points or switching capacitors) based on the learned policy. Reward Calculation: Compute the reward based on power loss reduction and voltage regulation. Policy Update: The agent updates its policy using a DRL algorithm (such as PPO).
Step III: Particle Swarm Optimization (PSO) Process:

- Initialize Swarm: Generate a swarm of particles representing potential solutions (control variables such as voltage levels or generator outputs).
- Velocity and Position Update: Each particle's velocity and position are updated based on its best-found solution and the global best.
- Evaluate Objective Function: For each particle, calculate the power losses and ensure constraints (e.g., voltage limits) are satisfied.
- Update Swarm Bests: Update the best positions and velocities of the particles.
- Check for Convergence: If the swarm has converged, the final control solution is determined.
- Final Solution: The optimal control actions are selected, minimizing power losses and maintaining voltage levels within desired limits.
- End: The algorithm terminates with optimized grid control actions.

2.2.5 Improved PSO:

Typical optimization for particle swarms. The intelligence method known as standard particle swarm optimization (PSO) is based on swarms [35]. First, a collection of uniformly distributed particles is initialized at random. Every particle represents a potential fix for the issue at hand. Next, the particles' own speed is modified based on its population extreme value (g^{best}) and individual extreme value (p^{best}). To follow the current optimal particle, all particles continuously adjust their position and speed. After continuous iteration, the optimal solution to the problem that satisfies the termination condition is obtained. The PSO algorithm's benefits include a straightforward structure, a small number of adjustable parameters, and quick search speed; nevertheless, it also has issues with premature convergence. Therefore, to keep the algorithm from prematurely convergent while simultaneously achieving rapid convergence, several enhancements are required to improve its local search capability and preserve population variety. The following are the primary improvement measures [36]:

Inertia Weight: To enhance the PSO algorithm's convergence performance, Shi and Eberhart

[31] added an inertia weight to the speed evolution equation:

$$v_i(t+1) = wv_i(t) + c_1r_1[p_i^{best} - x_i(t)] + c_2r_2[g^{best} - x_i(t)]$$

Where, w stand for the weight of inertia. Particles with inertial weight can travel over uncharted territory. The PSO algorithm's search efficiency and convergence accuracy can be enhanced by appropriately selecting the inertia weight, which balances local and global search [37]. The following are specific measures for improvement:

$$w = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} k$$

The maximum and minimum values of w w_{max} and w_{min} , typically take values of 0.9 and 0.4. The maximum number of iterations is $iter_{max}$. The current iteration is denoted by K .

Shrinkage Factor: Clerc introduced the shrinkage factor hypothesis in 1999. This approach can guarantee algorithm convergence by selecting w , c_1 , and c_2 in a suitable manner. The particle's velocity evolution equation is as follows when the shrinkage factor is added:

$$v_i(t+1) = x[wv_i(t) + c_1r_1[p_i^{best} - x_i(t)] + c_2r_2[g^{best} - x_i(t)]]$$

Among them, shrinkage factors are:

$$x = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$$

$$\varphi = c_1 + c_2$$

The particle velocity may be efficiently controlled by the shrinkage factor, which also strikes a good balance between local and global optimization. In addition, the flowchart of IPSO [38] is given in Figure 4:5.

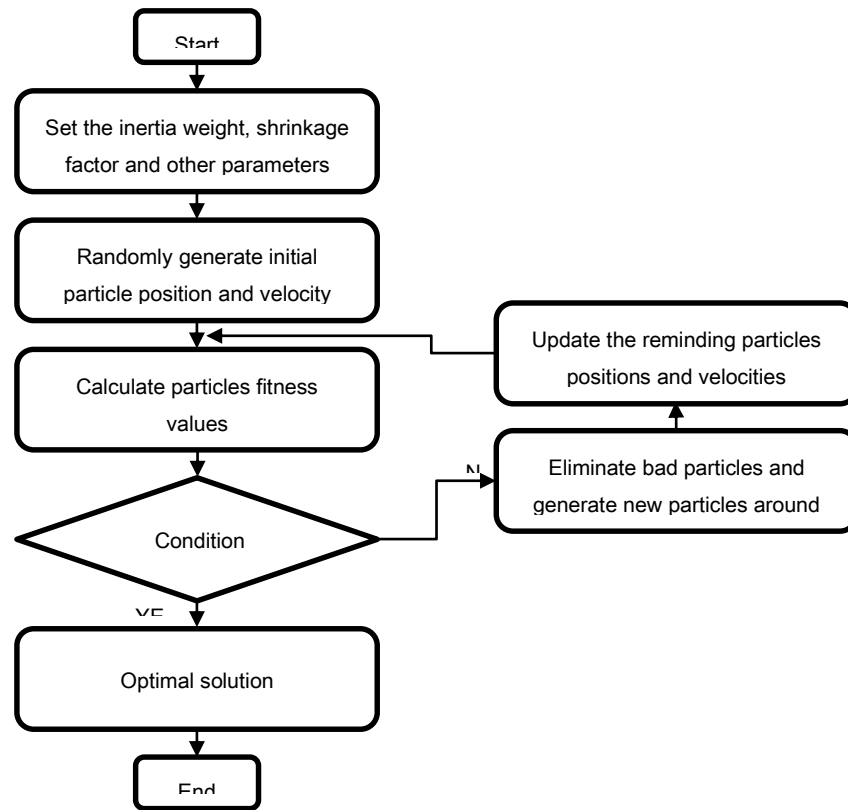


Fig 2:5 flow chart of IPSO algorithm

2.2.6 Multi-Objective Particle Swarm Optimization

The Multi-Objective Particle Swarm Optimization algorithm was introduced in [39] as an extension to the single objective PSO algorithm. The MOPSO algorithm handles multi objective optimization problems by using Pareto ranking scheme. To continuously replace a hard and fast Pareto best answers acquired with the aid of using MOPSO for the duration of iterations, this painting designs the historic Pareto best answer set and the worldwide Pareto optimal solution set during iterations with the help of archiving technology. Global Pareto most suitable answer set holds all Pareto most suitable answers generated for the duration of the modern-day iteration. If a population contains m particles, and each particle has N objective

function value, the global Pareto optimal solution set generated by each iteration is found by the following

1. Let $i=1$.
2. Compare particles x_i with particles x_j for all $j=1, 2, \dots, m$ and $j \neq i$.
3. If j exists so that particle x_j dominates x_i , the particle x_i is marked as the inferior solution.
4. If $i > m$, turn to 5. Otherwise, let $i=i+1$ and turn to 2.
5. Remove all marked answers, and the last answers represent the worldwide Pareto highest quality answer set of this iteration.

The very high convergence speed represents one of the main advantages provided by the PSO algorithm. Nevertheless, when multi-objective optimization problems are approached, the high convergence speed may turn into a major disadvantage as it may lead to a false Pareto front. The following improvements are implemented to overcome this drawback: crowding, mutation and dominance. Firstly, the crowding distance mechanism maintains the diversity of the non-dominated solutions by guiding the particles towards the least crowded area of the Pareto front. Secondly, the dominance strategy enforces a minimal distance between any two non-dominated solutions so that a wider spread of solutions along the Pareto front is assured. In addition, the flowchart of MOPSO [40] is given in Figure 4:6

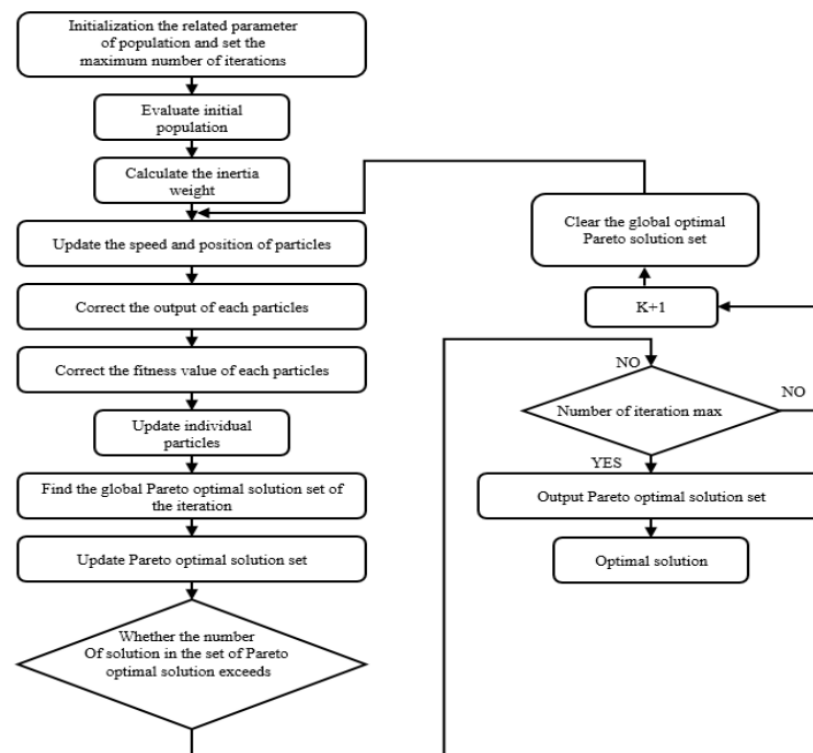


Fig 2.2: flow chart of MOPSO algorithm

2.2.7 Multi-Objective Wind-Driven Optimization Algorithm

The multi-objective energy optimization problems include clashing objects under equivalency and inequality constraints, which must be answered contemporaneously. In this

work, two clashing objects like power losses and voltage divagation must be optimized contemporaneously. MOWDO algorithm is grounded on position and haste conforming of 5 main functions Schaffer function, Kita function, Kursawe function, ZDT1 and ZDT4 functions. Also, the operating cost is a non-convex optimization problem. Thus, to avoid original minimum, we increase the probability of disquisition (global minimization) further than the probability of exploitation (original minimization) in the hunt space. The flyspeck converges to global minimization, and original minimization is avoided for operating cost optimization. MOWDO algorithm uses Pareto set species to find the stylish possible result. The MOWDO algorithm inflow map is shown in Fig. 4.7.

1. Schaffer function

In this function, limits of variables are and $[-10^3, 10^3]$, optimized result ranges (0, 2). Schaffer functions are:

$$f_1(k) = k^2, \quad f_2(k) = (k - 2)^2,$$

2. Kita function

Then, the limits of variables are (0, 7). Kita multi-objective functions are:

$$f_1(k_1, k_2) = -k_1^2 + k_2$$

And

$$f_2(k_1, k_2) = \frac{k_1}{2} + k_2 + 1$$

Subject to:

$$\frac{k_1}{6} + k_2 \leq \frac{13}{2}, \frac{k_1}{2} + \frac{15}{2}, 5k_1 + k_2 \leq 30$$

This characteristic is used to make use of strain effect.

3. Kursawe function

Here, the limits of variables are $[-5, 5]$. Kursawe multi-objective functions are:

$$f_1(k) = \sum_{j=1}^{N-1} (-10 \exp(-0.2 \sqrt{k_j^2 + k_{j+1}^2}))$$

$$f_2(k) = \sum_{j=1}^N |k_j|^{0.8} + 5 \sin(k_j^3)$$

Also, there are two functions which are used for assessing system model data, the ZDT1 and ZDT4 functions, independently.

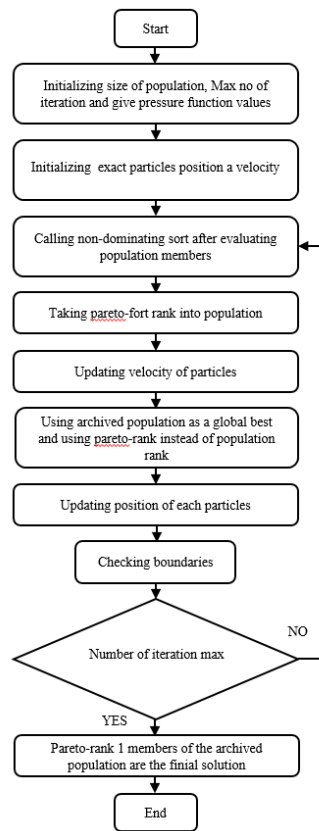


Fig 2.2: flow chart of MOWDO algorithm

2.2.8 Multi Objective Genetic Algorithm

MOGA algorithm [15] inflow map is shown in Fig. 48. MOGA use the dominated bracket of the GA population and at the same time maintain diversity in non-dominated results. The results, which are near to Pareto most excellent, the front is ranked same to at least one is proposed. These results are optimal results. Also, all other results are ranked consequently, grounded on their position. To find the rank of a result, the following equation is used:

$$R_k = 1 + N_k$$

where R_k indicates the rank of results and N_k represents that how numerous results are which can dominate the result k , if many results dominate, it means that the rank is advanced. To combine further than one ideal, the equation as follows:

$$F(X) = m_1 \cdot F_1(X) + \dots + m_j \cdot F_j(X) + \dots + m_N \cdot f_N(X)$$

where X is string of the rank, $F(X)$ is fitness function, $F_j(X)$ is j th objective function and m_1 is a constant weight which indicates object's function. The operating cost is non-convex optimization problem, thus, to avoid original minima, in the hunt space, we increase the probability of disquisition (global minimization) further than the probability of exploitation (original minimization). The flyspeck converges to global minimization and original minimization avoided for operating cost optimization. Now then we banded how to assign fitness values to MOGA. Assign health values are calculated as follows:

Step 1: Choose σ_{share} , which is a constant variable, denotes how important distance is considered between two solutions. However, it has a lower value also we say that the results are near, If σ_{share} .

Step 2: Cipher the number of results N_k and rank of result R_k as shown in Eq.

$$F_k = N_k - \sum_{k=1}^{R_k+1} \mu(k) - 0.5[\mu(R_k) - 1]$$

Step 3: If $k \propto N$, $k = k + 1$ also go to step 1. Else, go to step 4 shown in MOGA inflow map.

Step 4: Identify maximum rank R_k Eq. will give average fitness to each result k . Where N_k is total number of results, μk is number of results of the rank R_k , $()$ are the number of results in the current rank.

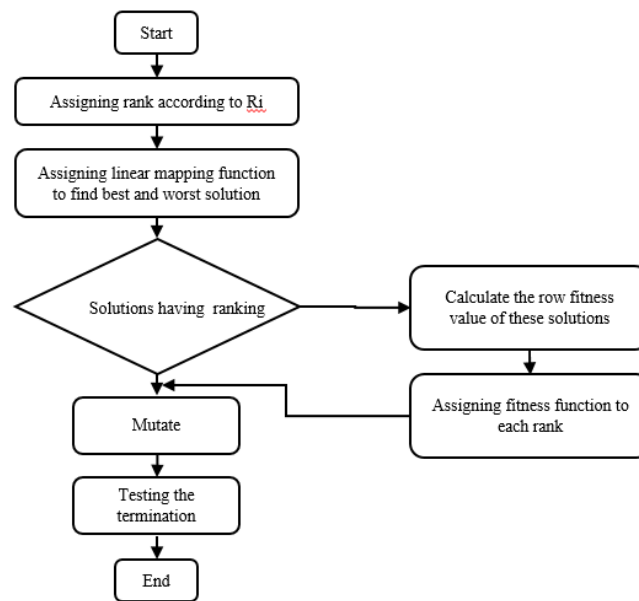


Fig 2.2: flow chart of MOGA algorithm

The DRL agent uses a learning rate of 0.0003, with an actor-critic architecture featuring 2 hidden layers (64 neurons each) for the actor and 2 hidden layers (128 neurons each) for the critic. The reward function combines power loss minimization, voltage deviation reduction, and convergence efficacy, formulated as $R = -(\text{power loss} + \lambda_1 * \text{voltage deviation} + \lambda_2 * \text{convergence time})$, where λ_1 and λ_2 are weighting factors.

2.3 Test System:

2.3.1 Case study:

According to recent energy optimization studies, by optimizing power generation and utilization, energy consumption can be lowered to 25–35% without requiring changes to the current system infrastructure. The distribution system operator oversees using the transmission network's high degree of efficiency and minimal losses to deliver electricity to the end user's load in the power system network. Renewable Energy sources are alternative power sources that give DSO new ways to satisfy end customers' demands and supply them with affordable power while minimizing operating costs, pollution, and loss of load expectation. As a result, lowering load shedding and improving power system network dependability. Conversely, DSO uses DSM techniques to carry out optimal load control. This paper is evaluating the best AI-Optimization Technique among these optimization techniques (PSO, GA, MOGA, DRL-PSO, DE, Improved PSO, MOPSO, and MOWDO) in respect of voltage deviation, power loss and running time.

With a combination of commercial, industrial, and residential loads, the 33-bus test system is intended to replicate a typical distribution test system. It is therefore a perfect testbed for assessing the performance of distribution systems. The test system, which has 33 buses, is both enough large to replicate the complexities of a real-world distribution system and sufficiently small to maintain computational feasibility. The framework of the IEEE 33-bus test system is straight forward and well-defined, which enables comprehension and modification. This test system is ideal for analysing smart grid concepts such as demand response, energy storage and distribution generation. This system is also useful in the way that it consists of minimum number of generators. Power flow, voltage management, frequency regulation, protection, economic dispatch, and control coordination become more difficult when a power system consists of several generators. To guarantee dependable and effective system operation, sophisticated modelling, simulation, and optimization techniques are needed. So, this system is neither too complex nor simple.

5.1.1 IEEE Bus Line Data

No.	From Bus	To Bus	Resistance (R)	Reactance (X)
1	1	2	0.0922	0.047
2	2	3	0.493	0.2511
3	3	4	0.366	0.1864
4	4	5	0.3811	0.1941
5	5	6	0.819	0.707
6	6	7	0.1872	0.6188
7	7	8	0.7114	0.2351
8	8	9	1.03	0.74
9	9	10	1.044	0.74
10	10	11	0.1966	0.065
11	11	12	0.3744	0.1238
12	12	13	1.468	1.155
13	13	14	0.5416	0.7129
14	14	15	0.591	0.526
15	15	16	0.7463	0.545
16	16	17	1.289	1.721

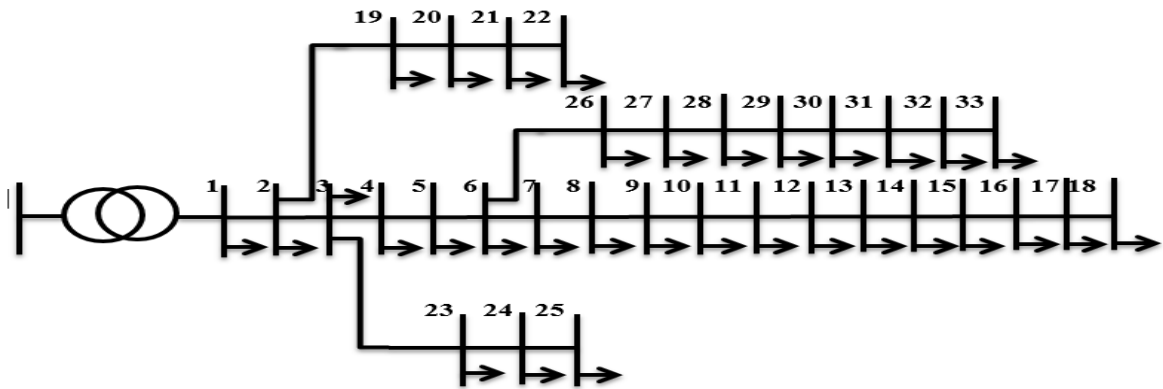
17	17	18	0.732	0.574
18	2	19	0.264	0.2565
19	19	20	1.5042	1.3554
20	20	21	0.4095	0.4784
21	21	22	0.7089	0.9373
22	3	23	0.4512	0.3083
23	23	24	0.898	0.7091
24	24	25	0.896	0.7011
25	6	26	0.203	0.1034
26	26	27	0.2842	0.1447
27	27	28	1.059	0.9337
28	28	29	0.8042	0.7006
29	29	30	0.5075	0.2585
30	30	31	0.9744	0.963
31	31	32	0.3105	0.3619
32	32	33	0.341	0.5302

5.1.2 IEEE Bus Load Data

Bus No.	Real Power (kW)	Reactive Power (kVAR)
1	0	0
2	100	60
3	90	40
4	120	80
5	60	30
6	60	20
7	200	100
8	200	100
9	60	20
10	60	20
11	45	30
12	60	35
13	60	35
14	120	80
15	60	10
16	60	20
17	60	20
18	90	40
19	90	40
20	90	40
21	90	40
22	90	40
23	90	50
24	420	200
25	420	200
26	60	25
27	60	25
28	60	20
29	120	70

30	200	600
31	150	70
32	210	100
33	60	40

This paper considers the IEEE 33-bus System as test-system, consists of 33 buses, 32 distributed lines and one generator and data for system parameters is obtained from IEEE Standard Benchmark Dataset. Given figure 5:1 shows single-line-diagram of test-system.



2.3: single line diagram of IEEE 33-BUS distribution network

3 Results and Discussion

1.1. Simulation Setup:

MATLAB is a high-level programming language and software environment created especially for numerical computing, data analysis, and visualization. Engineers, scientists, and researchers favor MATLAB because it was developed to carry out intricate mathematical operations, simulate systems, and create models.

A MATLAB library called Tensor Flow was created to address optimum power flow (OPF) issues in Electrical System especially in power system. For modelling and resolving intricate power flow optimization issues, such as power flow (AC and DC), convex relaxations, and bespoke objective functions, it offers a versatile and effective framework. Tensor Flow allows users to plan and construct power grids, optimize power system operations, and assess the effects of energy storage and renewable energy sources. Tensor Flow is a useful tool for power system researchers, engineers, and analysts since it allows users to rapidly create and evaluate power flow optimization strategies by utilizing MATLAB's numerical computation capabilities.

Simulation of several algorithms is the purpose of this paper. In this way, MATLAB is being used. Tensor flow is employed because algorithms are used for optimization. For each of the algorithms mentioned above, a test system that is mentioned above is simulated, and the best outcomes are compared in terms of frequency deviation, power loss and algorithm running time.

1.2. Results & Comparative analysis:

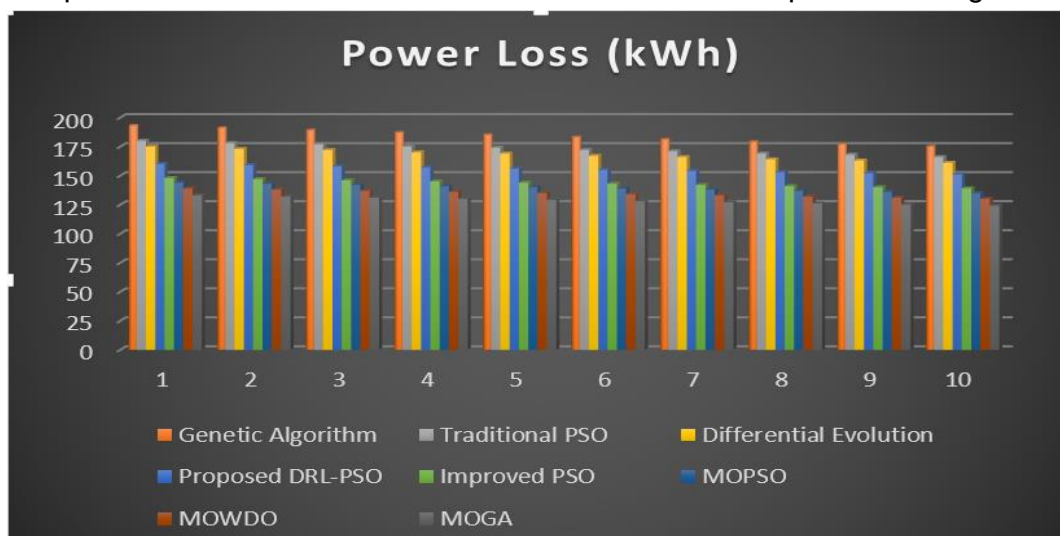
1.2.1. Power Loss

The graph 6.1 illustrates the Total Power Losses vs. Iterations for various optimization algorithms applied to a power system. The x-axis represents the number of iterations, while the y-axis shows total power losses in kilowatts (kW). The objective of each algorithm is to minimize power losses as iterations progress. Among the methods analysed, Multi-Objective Genetic Algorithm (MOGA) demonstrates the best performance, achieving the lowest power losses (~130 kW) and the highest percentage reduction of approximately **30–35%**, making it the most effective method. Similarly, Multi-Objective Wind Driven Optimization (MOWDO), Multi-Objective PSO (MOPSO), and Improved PSO exhibit strong optimization capabilities, significantly reducing power losses by **26–32%**. The Proposed DRL-PSO method also outperforms traditional techniques, achieving a reduction of **22–26%**. In contrast, Differential Evolution (DE), Traditional PSO, and Genetic Algorithm (GA) provide moderate improvements, with power loss reductions ranging from **12–20%**. Linear Programming, which starts with the highest power loss (~220 kW), shows the least effectiveness, achieving only an **8–10%** reduction. Overall, the results highlight the superiority of AI-driven and hybrid optimization techniques in enhancing power system efficiency by reducing losses more effectively than conventional approaches.

Table 1.2-1 Performance of optimization methods in power loss reduction.

Methods	Iterations	Power Loss Range (Kw)
Genetic Algorithm	1-10	193-175
Particle Swarm Optimization	1-10	180-166
Differential Evolution	1-10	175-161
Proposed DRL-PSO	1-10	160-151
Improved PSO	1-10	148-139
Multi Objective PSO	1-10	144-135
Multi Objective WDO	1-10	139-130
Multi Objective GA	1-10	133-124

Graph 3-1 Total Power Losses vs. Iterations for various optimization algorithms



The graph presented provides a comparative analysis of various optimization algorithms applied to power system optimization, focusing on total power losses, voltage deviation, and convergence time. The first bar chart illustrates total power losses in kilowatts (kW), where the primary objective is to minimize losses for improved system efficiency. Among the methods analysed, Linear Programming (LP) exhibits the highest power losses, making it the least effective approach. Genetic Algorithm (GA), Traditional PSO, and Differential Evolution (DE) achieve moderate reductions, with power loss minimization ranging from 10% to 18%. In contrast, the Proposed DRL-PSO and Improved PSO demonstrate significant improvements, reducing power losses by approximately 22–28%. The best-performing methods, including Multi-Objective PSO (MOPSO), Multi-Objective Wind Driven Optimization (MOWDO), and Multi-Objective Genetic Algorithm (MOGA), achieve the lowest total losses (~140–160 kW), with an impressive reduction of 30–35%, making them the most efficient optimization techniques.

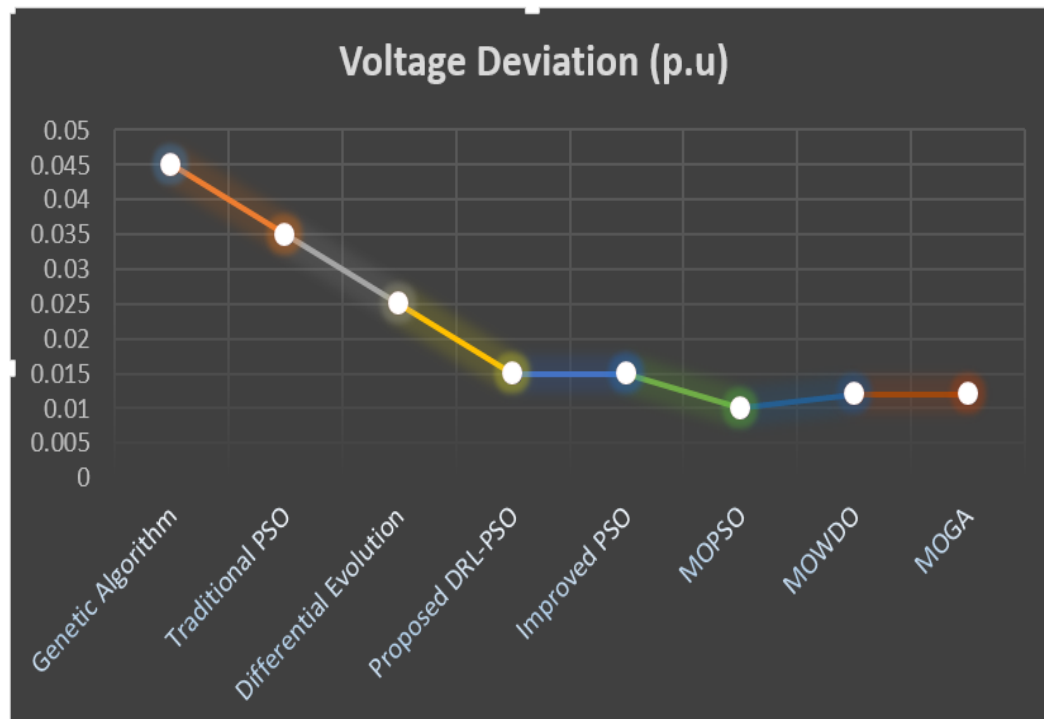
1.2.2. Voltage deviation:

The second bar chart compares voltage deviation across different methods, where lower values indicate better voltage stability. Linear Programming results in the highest deviation (~0.09 p.u.), highlighting its inefficiency in maintaining voltage levels. GA, Traditional PSO, and DE show moderate improvements, achieving reductions in the range of 12–20%. Meanwhile, the Proposed DRL-PSO and Improved PSO further optimize voltage profiles, bringing deviations below 0.02 p.u. The best results are again observed with MOPSO, MOWDO, and MOGA, which achieve the lowest deviations (~0.01–0.015 p.u.), ensuring optimal voltage stability.

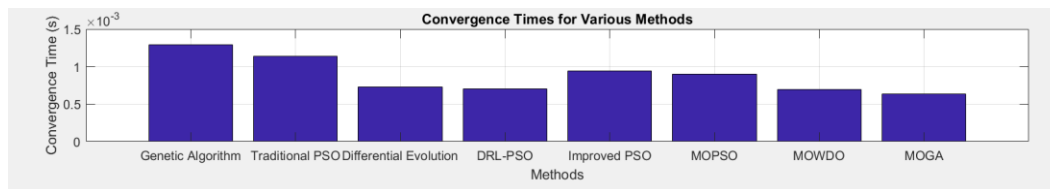
The third bar chart illustrates convergence time, representing how quickly each algorithm reaches an optimal solution. Most methods show similar convergence speeds, except for the Proposed DRL-PSO, which requires more time due to the complexity of deep reinforcement learning. However, MOPSO, MOWDO, and MOGA exhibit the fastest convergence times while maintaining superior optimization performance, making them ideal for real-time applications.

Table 1.2-2 Performance of Optimization Methods in Voltage Deviation.

Methods	Iterations	Voltage Deviation (P.U)
Genetic Algorithm	1-10	0.045
Particle Swarm Optimization	1-10	0.035
Differential Evolution	1-10	0.025
Proposed DRL-PSO	1-10	0.015
Improved PSO	1-10	0.015
Multi Objective PSO	1-10	0.010
Multi Objective WDO	1-10	0.012
Multi Objective GA	1-10	0.012



Overall, the results highlight the superiority of AI-driven and hybrid optimization techniques in power system optimization. MOPSO, MOWDO, and MOGA consistently achieve the best performance, minimizing power losses, reducing voltage deviation, and ensuring fast convergence. The Proposed DRL-PSO and Improved PSO also demonstrate strong optimization capabilities but require longer computational times. In contrast, Linear Programming proves to be the least effective approach, struggling with both high-power losses and voltage deviations. These findings emphasize the importance of advanced optimization techniques in enhancing power system efficiency and stability.



4 Conclusion

This research is based on dual purpose: Optimizing grid losses using advanced AI optimization techniques and providing a comparative analysis of their performances. The results show that AI-driven method and hybrid optimization methods, particularly MOGA, MOWDO and MOPSO, significantly reduce power losses of about 30-35% while ensuring minimal voltage deviations (~0.01-0.015p.u.) and required less time for convergence. The proposed DRL-PSO and improved PSO also exhibit strong optimization potential, though at the cost of higher convergence time. At the same time, comparative analysis also presents the limitations of conventional methods like Linear Programming, Traditional PSO and Genetic Algorithm, which achieve only moderate improvements in power loss reduction and voltage stability. These findings lead to the importance of AI-driven and hybrid techniques for enhancing power system efficiency, offering best solutions over traditional methods. By

optimizing power losses while evaluating the effectiveness of differential algorithms, this study contributes valuable insights toward developing more intelligent, stable and energy-efficient power networks. Expanding scope of optimization to include economic and environmental factors such as cost and CO₂ emission with Development of AI-based methods that can dynamically adapt to grid fluctuations in real-time scenarios can be potential research topics in future.

5 Reference

1. S.-M. Razavi, H.-R. Momeni, M.-R. Haghifam, and S. Bolouki, "Multi-Objective Optimization of Distribution Networks via Daily Reconfiguration," *IEEE Transactions on Power Delivery*, vol. 37, no. 2, pp. 775–785, Apr. 2022, doi: 10.1109/TPWRD.2021.3070796.
2. United Nations, "The SDGs: 17 Goals to Transform Our World," United Nations.
3. D. Mourtzis and J. Angelopoulos, "Reactive Power Optimization Based on the Application of an Improved Particle Swarm Optimization Algorithm," *Machines*, vol. 11, no. 7, p. 724, Jul. 2023, doi: 10.3390/machines11070724.
4. M. A.Wahab ALI, T. A. H. Alghamdi, M. Alenezi, and M. Hatatah, "DRL-PSO Powered Optimization for Efficiency Enhancement and Reliability Improvement in Smart Grids," Nov. 04, 2024. doi: 10.20944/preprints202411.0132.v1.
5. T. Xie, G. Zhang, J. Xie, and Y. Liu, "An improved particle swarm optimization algorithm for reactive power optimization," in *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*, Toronto, Canada: IEEE, Dec. 2013, pp. 489–493. doi: 10.1109/IMSNA.2013.6743322.
6. S.-M. Razavi, H.-R. Momeni, M.-R. Haghifam, and S. Bolouki, "Multi-Objective Optimization of Distribution Networks via Daily Reconfiguration," *IEEE Transactions on Power Delivery*, vol. 37, no. 2, pp. 775–785, Apr. 2022, doi: 10.1109/TPWRD.2021.3070796.
7. C. Fu, J. Liu, J. Zeng, and M. Ma, "Adaptive Reactive Power Optimization in Offshore Wind Farms Based on an Improved Particle Swarm Algorithm," *Electronics (Basel)*, vol. 13, no. 9, p. 1637, Apr. 2024, doi: 10.3390/electronics13091637.
8. Z. Li and J. Xiong, "Reactive Power Optimization in Distribution Networks of New Power Systems Based on Multi-Objective Particle Swarm Optimization," *Energies (Basel)*, vol. 17, no. 10, p. 2316, May 2024, doi: 10.3390/en17102316.
9. D. Carrascal, P. Bartolomé, E. Rojas, D. Lopez-Pajares, N. Manso, and J. Diaz-Fuentes, "Fault Prediction and Reconfiguration Optimization in Smart Grids: AI-Driven Approach," *Future Internet*, vol. 16, no. 11, p. 428, Nov. 2024, doi: 10.3390/fi16110428.
10. "Application of Optimization Techniques in the Power System Control," *ACTA POLYTECHNICA HUNGARICA*, vol. 10, no. 5, Sep. 2013, doi: 10.12700/APH.10.05.2013.5.14.
11. G. A. M., "SMART GRID COST OPTIMIZATION USING GENETIC ALGORITHM," *Int. J. Res. Eng. Technol.*, vol. 03, no. 19, pp. 282–287, May 2014, doi: 10.15623/ijret.2014.0319051.
12. M. G. N. K. S. S. Kumar, "Load Shifting Technique on 24Hour Basis for a Smart-Grid to Reduce Cost and Peak Demand Using Particle Swarm Optimization," *International Research Journal of Engineering and Technology (IRJET)*, vol. 04, no. 10, pp. 1180–1185, Oct. 2017.
13. K. M. Tan, V. K. Ramachandaramurthy, and J. Y. Yong, "Integration of electric vehicles in smart grid: A review on vehicle to grid technologies and optimization techniques," *Renewable and Sustainable Energy Reviews*, vol. 53, pp. 720–732, Jan. 2016, doi: 10.1016/j.rser.2015.09.012.
14. S. Salinas, M. Li, and P. Li, "Multi-Objective Optimal Energy Consumption Scheduling in Smart Grids," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 341–348, Mar. 2013, doi: 10.1109/TSG.2012.2214068.
15. K. Ullah, G. Hafeez, I. Khan, S. Jan, and N. Javaid, "A multi-objective energy optimization in smart grid with high penetration of renewable energy sources," *Appl. Energy*, vol. 299, p. 117104, Oct. 2021, doi: 10.1016/j.apenergy.2021.117104.
16. C. Roy and D. K. Das, "A hybrid genetic algorithm (GA)–particle swarm optimization (PSO)

- algorithm for demand side management in smart grid considering wind power for cost optimization,” *Sādhana*, vol. 46, no. 2, p. 101, Jun. 2021, doi: 10.1007/s12046-021-01626-z.
17. S. S. Reddy, V. Sandeep, and C.-M. Jung, “Review of stochastic optimization methods for smart grid,” *Frontiers in Energy*, vol. 11, no. 2, pp. 197–209, Jun. 2017, doi: 10.1007/s11708-017-0457-7.
 18. C. Roy, D. K. Das, and A. Srivastava, “Particle Swarm Optimization based Cost Optimization for Demand Side Management in Smart Grid,” in *2019 International Conference on Electrical, Electronics and Computer Engineering (UPCON)*, IEEE, Nov. 2019, pp. 1–6. doi: 10.1109/UPCON47278.2019.8980149.
 19. H. R. S. R. P. B. D. W. S. Ritter, “New Approaches for Smart Grid Requirements: Grid Protection and Optimization of Distribution Grid Operation,” Detroit, MI, USA: IEEE, Jul. 2011, pp. 1–7.
 20. G. D. P. C. Ramaswamy, “Relevance of Voltage Control, Grid Reconfiguration and Adaptive Protection in Smart Grids and Genetic Algorithm as an Optimization Tool in Achieving their Control Objectives,” IEEE, Apr. 2011.
 21. P. Pezzini, O. Gomis-Bellmunt, and A. Sudrià-Andreu, “Optimization techniques to improve energy efficiency in power systems,” *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 2028–2041, May 2011, doi: 10.1016/j.rser.2011.01.009.
 22. J. Bryson and P. D. Gallagher, “NIST Special Publication 1108R2: Framework and Roadmap for Smart Grid Interoperability Standards, Release 2.0,” 2012.
 23. J. Li, R. Xiong, Q. Yang, F. Liang, M. Zhang, and W. Yuan, “Design/test of a hybrid energy storage system for primary frequency control using a dynamic droop method in an isolated microgrid power system,” *Appl. Energy*, vol. 201, pp. 257–269, Sep. 2017, doi: 10.1016/j.apenergy.2016.10.066.
 24. P. S. Indu and M. V. Jayan, “Frequency regulation of an isolated hybrid power system with Superconducting Magnetic Energy Storage,” in *2015 International Conference on Power, Instrumentation, Control and Computing (PICC)*, IEEE, Dec. 2015, pp. 1–6. doi: 10.1109/PICC.2015.7455752.
 25. I. Patrao, E. Figueres, G. Garcerá, and R. González-Medina, “Microgrid architectures for low voltage distributed generation,” *Renewable and Sustainable Energy Reviews*, vol. 43, pp. 415–424, Mar. 2015, doi: 10.1016/j.rser.2014.11.054.
 26. O. Palizban and K. Kauhaniemi, “Energy storage systems in modern grids—Matrix of technologies and applications,” *J. Energy Storage*, vol. 6, pp. 248–259, May 2016, doi: 10.1016/j.est.2016.02.001.
 27. A. Mohamed Imran, M. Kowsalya, and D. P. Kothari, “A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks,” *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 461–472, Dec. 2014, doi: 10.1016/j.ijepes.2014.06.011.
 28. J. Modarresi, E. Gholipour, and A. Khodabakhshian, “A comprehensive review of the voltage stability indices,” *Renewable and Sustainable Energy Reviews*, vol. 63, pp. 1–12, Sep. 2016, doi: 10.1016/j.rser.2016.05.010.
 29. K. E. Adetunji, I. W. Hofsajer, A. M. Abu-Mahfouz, and L. Cheng, “A Review of Metaheuristic Techniques for Optimal Integration of Electrical Units in Distribution Networks,” *IEEE Access*, vol. 9, pp. 5046–5068, 2021, doi: 10.1109/ACCESS.2020.3048438.
 30. M. R. Islam, H. H. Lu, M. J. Hossain, and L. Li, “A Comparison of Performance of GA, PSO and Differential Evolution Algorithms for Dynamic Phase Reconfiguration Technology of a Smart Grid,” in *2019 IEEE Congress on Evolutionary Computation (CEC)*, IEEE, Jun. 2019, pp. 858–865. doi: 10.1109/CEC.2019.8790357.
 31. D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” *Soft comput.*, vol. 22, no. 2, pp. 387–408, Jan. 2018, doi: 10.1007/s00500-016-2474-6.
 32. A. G. Gad, “Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review,” *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.
 33. R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–

- 359, Dec. 1997, doi: 10.1023/A:1008202821328.
34. V. V. Sudhakar Angatha, K. Chandram, and A. J. Laxmi, "Bidding Strategy in Deregulated Power Market Using Differential Evolution Algorithm," *Journal of Power and Energy Engineering*, vol. 03, no. 11, pp. 37–46, 2015, doi: 10.4236/jpee.2015.311004.
 35. H. Zhang, G. Li, and S. Wang, "Optimization dispatching of isolated island microgrid based on improved particle swarm optimization algorithm," *Energy Reports*, vol. 8, pp. 420–428, Dec. 2022, doi: 10.1016/j.egyr.2022.10.199.
 36. T. Wang; H. Jiang; K. Xu; G. Li, "Reactive Power Optimization of Power System based on Improved Particle Swarm Optimization," in *International Conference on Development and Research in Power Technology (DRPT)*, 2011, pp. 606–609.
 37. M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002, doi: 10.1109/4235.985692.
 38. Y. Shi; R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, New York, NY, USA: Springer-Verlag, 1998, pp. 591–600.
 39. C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, IEEE, 2002, pp. 1051–1056 vol.2. doi: 10.1109/CEC.2002.1004388.
 40. C. Nartey et al., "Blockchain-IoT peer device storage optimization using an advanced time-variant multi-objective particle swarm optimization algorithm," *EURASIP J. Wirel. Commun. Netw.*, vol. 2022, no. 1, p. 5, Dec. 2022, doi: 10.1186/s13638-021-02074-3.