JOURNAL OF BASIC SCIENCE AND ENGINEERING

Comparative Analysis of Chameleon Swarm and Particle Swarm Optimization to Optimize Energy Scheduling in Power System

Dhanesh S Patil
Research Scholar
KBC North Maharashtra University Jalgaon
Department of Electrical Engineering,
G H Raisoni College of Engineering and Management, Jalgaon

Dr. Vijay M Deshmukh
Research Guide
Department of Electronics and Telecommunication Engineering
S. S. B. T.'s, College of Engineering and Technology, Bambhori, Jalgaon

DOI: https://doie.org/10.10399/JBSE.2025540584

Abstract
Energy scheduling is a key function in optimizing the generation, transmission, and consumption of power, especially with the growing adoption of renewable energy sources in today's power systems. This article is a comparative analysis of two popular metaheuristic algorithms—Chameleon Swarm Algorithm (CSA) and Particle Swarm Optimization (PSO) for optimized energy scheduling. Overall consensus of this research is on the observation of the convergence patterns of CSA and PSO when used on energy scheduling problems, as in minimizing generation costs and exploiting the usage of renewable energy. The performance of the two algorithms is tested through numerical simulations on the basis of their convergence rate, solution quality, and resilience under different system scenarios. The findings highlight the variation between each algorithm to arrive at the optimum resolution, special consideration to efficiency of algorithms in coping with intricate power system constraints. The research proves that although CSA and PSO are efficient energy scheduling solutions, they possess different kinds of convergence behaviors—CSA is more performance-oriented and quicker in convergence but reliability, while PSO is superior in detecting global optima with fewer steps. The findings of the study are valuable in terms of knowledge regarding all the algorithms' merits and shortcomings, and offering insight for determining the most proper optimization technique that should be implemented for energy scheduling in different applications of power systems.

Keywords: *Energy Scheduling, Chameleon Swarm Algorithm, Particle Swarm Optimization, Convergence Patterns, Renewable Energy Integration, Metaheuristic Algorithms, Power System Optimization, Global Optima, Convergence Speed, Solution Accuracy.*

1. Introduction:

The need for energy is continuously increasing due to the integration of different forms of energy, which has brought energy scheduling to the front of power system optimization. Energy scheduling is crucial because it determines the efficiency of energy generation, transmission, and consumption with minimal operational costs and maximal system reliability and sustainability. Renewable technologies like wind and solar introduce even more variability and intermittency into the system, further complicating the scheduling process. Adjusting to these sources of integration in many cases leads to a traditional optimization problem being transformed into a difficult energy scheduling problem where numerous intricate restrictions and multi-objective tasks need to be solved simultaneously. In contrast to traditional optimization methods, heuristic techniques give better opportunities to manage the complex tasks of energy scheduling. A few of the Chameleon Swarm Algorithm (CSA) and Particle Swarm Optimization (PSO) stay widely used heuristic methods. In PSO, the swarm behavior of individual particles is mimicked, where each particle moves to a position based on its private involvement and the experiences of additional particles in the group. In contrast, as chameleons adapt to their environment by changing their position and color, CSA approaches energy scheduling from a completely different perspective.

Although they are extensively used, PSO and CSA have varying convergence behaviors, which play a key role in their performance in energy scheduling applications. Convergence rate, solution quality, and stability are essential aspects when considering the applicability of these algorithms to practical problems. While both algorithms can produce optimal or near-optimal solutions, they have different convergence properties, and this influences their ability to solve complex power system constraints. The article exhibits a comparison between PSO and CSA as algorithms used to design energy schedules within power networks based on the performance of their convergence behavior. From a series of simulations, concert of both algorithms is tested in charge of convergence proportion, solution quality, and stability in various system operating conditions. In this study, specific tasks including minimizing generation costs, maximizing utilization of renewable power, and responding to different types of operational limits are considered. The results give insight into the strengths and shortcomings of every algorithm, and they can be used to guide selection of the best optimization method to use in energy scheduling in different power system applications.

2. Literature Review

This chapter summarizes current literature on optimization methods used for energy scheduling with an emphasis on metaheuristic algorithms such as Chameleon Swarm Optimization (CSA) and Particle Swarm Optimization (PSO).

2.1 Optimization in Energy Scheduling

Energy scheduling is an essential issue in power systems that focuses on maximizing the generation, transmission, and usage of energy with minimum costs and maintaining system reliability. Metaheuristic algorithms are extensively employed in energy scheduling problems because they can solve complicated, nonlinear, and large-scale optimization problems. Metaheuristic algorithms are naturally inspired and excel at finding near-optimal solutions in vast solution spaces.

2.2 Particle Swarm Optimization (PSO):

PSO is ace of the utmost trendy methods of optimization in energy scheduling. PSO models the social activity of particles (agents) within a swarm such that every particle updates its location based on both its own knowledge and its neighbouring particles' experience. PSO is particularly appropriate for non-linear optimization problems like those that appear in energy scheduling, where the objective function is extremely complex. PSO has been employed in power systems to reduce the generation cost, optimize the integration of renewable energies, and stabilize the load.

A. *Mathematical Model of PSO*

In PSO, every particle (candidate solution) in the bevy is a probable resolution in an n-dimensional space. Particles iteratively update their velocities and positions based on their own best and the best positions

i. *Particle Depiction:* Apiece particle in the bevy signifies a solution vector: $Xi = (xi1, xi2, \dots, xin)$ where Xi is the site of the i[th] particle in n-dimensional hunt planetary.
   Each particle also has a velocity vector: $Vi = (vi1, vi2, \dots, vin)$ which determines its movement in the search space.

ii. *Speed Inform Equation:* The rapidity of apiece particle is rationalized using the following equation:

$$V_i^{(t+1)} = w \cdot V_i^{(t)} + c_1 \cdot r_1 \cdot \left(PBest_i - X_i^{(t)}\right) + c_2 \cdot r_2 \cdot \left(GBest - X_i^{(t)}\right)$$

   where:
   - w = inertia weight (controls exploration vs. exploitation balance)
   - c_1, c_2= cognitive and social acceleration quantities
   - r_1, r_2 = arbitrary numbers in the series $[0,1]$
   - $PBest_i$ = own finest site of the ith particle
   - $GBest$ = universal finest site originates by the swarm

iii. *Site Update Equation:* Every particle uses the new velocity to update its position: $X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}$

iv. *Fitness Function:* The quality of each particle's position is assessed by the fitness function $f(X_i)$. The goal function for energy schedule optimization could be: min $f(X) = C_g + C_p + C_{res} - R$
   where, C_g is the cost of generating, C_p is the cost of power loss, C_{res}= cost of renewable energy integration, R is the revenue or energy savings.
   The constraints include:
   - Equation for power balance: $\sum P_{gen} = \sum P_{load} + P_{loss}$
   - Limits of the generator: $P_{min} \le P_{gen} \le P_{max}$
   - Limits on transmission capacity

B. *Algorithm of PSO*

Step 1: Initialization

- Set parameters: swarm size N, maximum iterations T_{max}, apathy heaviness w, cognitive coefficient c_1, and communal coefficient c_2.
- Set the search space's coordinates X_i and velocities V_i at random.
- Analyze each particle's fitness function, $f(X_i)$.
- Initial personal bests $PBest_i = X_i$ should be set.
- Out of all best $PBest_i$ find the global greatest $GBest$

Step 2: Iterative Update Process

For each iteration $t = 1$to T_{max}

- Bring up-to-date each particle's velocity: $V_i^{(t+1)} = w \cdot V_i^{(t)} + c_1 \cdot r_1 \cdot \left(PBest_i - X_i^{(t)}\right) + c_2 \cdot r_2 \cdot \left(GBest - X_i^{(t)}\right)$
- Bring up-to-date each particle's location: $X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}$
- Assess the purpose of fitness $f(X_i^{(t+1)})$
- Revise personal bests: If $f\left(X_i^{(t+1)}\right) < f(PBest_i)$ set $PBest_i = X_i^{(t+1)}$
- Bring up-to-date universal best: $if\ f(X_i^{(t+1)}) < f(GBest)$, sets $GBest = X_i^{(t+1)}$

Step 3: Convergence and Termination

The method should be terminated whenever the ultimate number of iterations is achieved or if the halting condition (such as no improvement in GBest is satisfied. The best option for energy scheduling is output GBest.

C. Flowchart of PSO

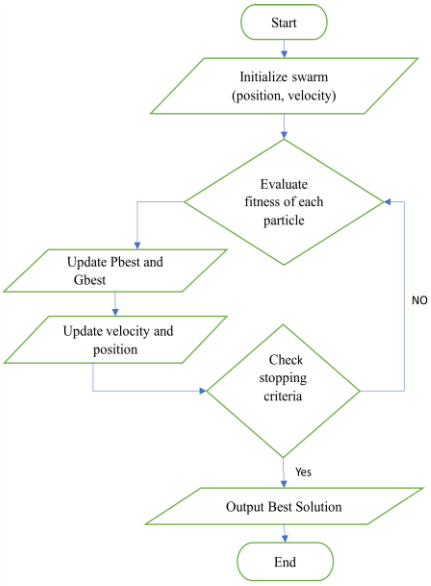

Figure 1: Flowchart of PSO

2.3 Chameleon Swarm Algorithm (CSA):

A relatively current bio-inspired optimization technique called Chameleon Swarm Optimization (CSA) imitates the adaptive hunting style of chameleons. It is extremely useful for uses such as calculating the capacity of storage units based on load, especially in terms of charging and discharging. Here, the ability of storage tanks to accommodate variable loads can be maximized. CSA, proposed by Braik in 2021, is a innovative metaheuristic procedure designed aimed at unconstrained optimization glitches. It is founded on dynamic search chameleon behaviour and is highly efficient in function optimization problems. The algorithm borrows its inspiration from the interesting behaviour of chameleons as they forage for food in different habitats like trees, deserts, and swamps. Chameleons are a peculiar sort of animals with special capability to alteration colour and camouflage themselves, allowing them to adapt to changing environments like lowlands, mountains, deserts, and semi-deserts. Their feeding mechanism has multiple steps: locating prey with their good eyesight, chasing the prey, and eventually catching it. The algorithm embeds this adaptability and applies it to build an optimization mechanism that is effective in solving intricate problems. The scientific replicas and the stepwise procedure of CSA are described in the subsequent sections.

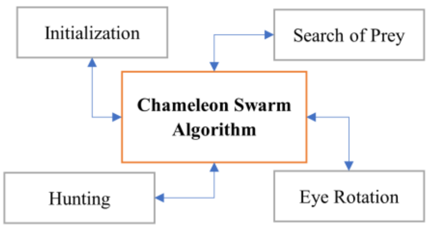

Figure 2: Scientific Model of CSA

A. Scientific Model of PSO

1. *Initiation and Execution Assessment:* Chameleon Swarm Algorithm (CSA) is a populace-built metaheuristics starts with a random initial population to be optimized. A populace of n chameleons is dispersed in a d-spatial hunt arena and each chameleon is a potential solution. Every chameleon at time t has a position equal to:

$$y_t^i = [y_{(t,1)}^i, y_{(t,2)}^i, \dots, y_{(t,d)}^i]$$

Here, "i" signifies the quantity of repetitions, $y_{(t,d)}^i$ signifies the location of the chameleon.

The first population is created by: $y^i = l_j + r(u_j - l_j)$, where $l_j$and u_j are the lower and upper limits of the hunt arena, and r is a random number between 0 and 1. The fitness of each solution is determined on the base of the impartial function.

2. *Exploration of Target:* Chameleons update positions according to a movement policy:

$$y_{(t+1)}^{(i,j)} = \begin{cases} y_t^{(i,j)} + \mu(u^j - l^j)r_3 + l_b^j \mathrm{sgn}(\mathrm{rand} - 0.5)r_1, & r_1 < P_p \\ y_t^{(i,j)} + P_1\left(P_t^{(i,j)} - G_t^j\right)r_2 + P_2\left(G_t^j - y_t^{(i,j)}\right)r_1, & r_1 \geq P_p \end{cases}$$

where P_1 and P_2 manage exploration, r_1, r_2, r_3 are arbitrary between 0 and 1, P_p is the probability of sleuthing prey, and μ reductions with iterations.

3. *Spin of the Chameleon's Eyes:* Chameleons employ eye revolution to find prey within 360°. The procedure includes:
   - The first position is the centre of gravity.
   - A spin matrix represents the position of the prey.
   - The chameleon's position is updated through the rotation matrix.
   - Chameleons go back to their starting points after the update
4. *Hunting Prey:* When prey is within reach, the nearest chameleon (best solution) strikes with its tongue, which can stretch twice its length to search a greater area. The velocity of the tongue is represented as:

$$v_{(t+1)}^{(i,j)} = \omega v_t^{(i,j)} + c_1\left(G_t^j - y_t^{(i,j)}\right) + c_2\left(P_t^{(i,j)} - y_t^{(i,j)}\right)r_2$$

where $v_{(t+1)}^{(i,j)}$ is the new velocity, $v_t^{(i,j)}$is the current velocity, and $c_1, c_2$are acceleration constants.

B. Algorithm of CSA

Step 1: Initialization

- Set parameters: population size N, maximum iterations T_{max}, control coefficients $\alpha, \beta, \gamma, \delta$.
- Initialize positions X_i randomly in the search space.
- Evaluate the strength task $f(X_i)$for all chameleons.
- Set initial best solution X_{best}

Step 2: Iterative Update Process

For each iteration $t = 1$ to T_{max}

1. Compute the social interaction component: $S_i^{(t)} = \alpha\left(X_{best} - X_i^{(t)}\right) + \beta\left(X_{rand} - X_i^{(t)}\right)$
2. Compute the environmental adaptation component: $E_i^{(t)} = \gamma \cdot \left(X_{rand} - X_i^{(t)}\right) + \delta \cdot R$
3. Update position of each chameleon: $X_i^{(t+1)} = X_i^{(t)} + S_i^{(t)} + E_i^{(t)}$
4. Evaluate the new fitness $f(X_i^{(t+1)})$.
5. Update best solution: If $f\left(X_i^{(t+1)}\right) < f(X_{best})$, set $X_{best} = X_i^{(t+1)}$

Step 3: Convergence and Termination

- If maximum iterations are reached or convergence criteria are met, terminate the algorithm.
- Output X_{best} as the optimized energy scheduling solution.

C. Flowchart of CSA

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

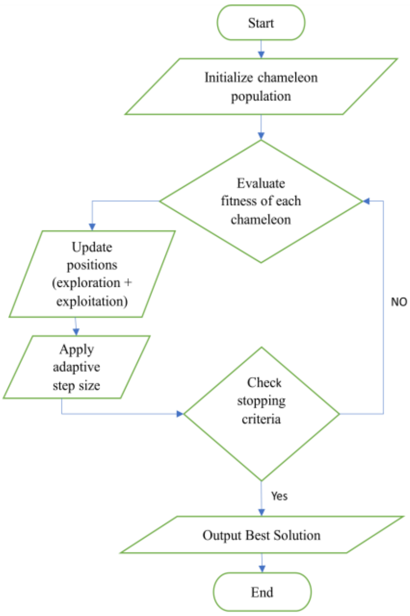

Figure 3: Flowchart of PSO

2.4 Research Gaps and Motivation

Although CSA has demonstrated good performance, little work has been done on its real-time energy scheduling application. Most research involves theoretical applications or small-scale simulations with gaps in actual deployment. This research seeks to:

- Compare the direct performance of CSA and PSO in energy scheduling.
- Compare their efficiency in cost optimization and renewable energy integration under different load conditions.
- Compare their computational efficiency and stability under different problem constraints.
- Offer insights into parameter tuning approaches for the two algorithms to improve their applicability in practice.

2.5 Problems in Optimized Scheduling of energy

Optimal energy scheduling problems are at the heart of optimal operation and control of contemporary power systems, particularly thru the lofty penetration of RES. These problems concern finding the most efficient and economic means of assigning energy resources among generation, storage, and distribution, while ensuring system stability and satisfying demand in real-time. Finding the best way to schedule energy use is crucial for reducing operational costs, enhancing the reliability of our power grid, and being kinder to the environment. This is particularly true now that many power grids are starting to use renewable energy sources like wind and solar, which can sometimes be a little unpredictable in terms of their energy output.

a. *Minimizing Generation Costs*: In order to reduce the expenses of generating electricity, we must devise means of reducing the costs of operating power plants while ensuring there is sufficient electricity for all. Coal, gas, or nuclear power plants typically have high construction and operational costs. Conversely, renewable energy sources like wind, solar, and hydropower generally cost less to run. However, they depend on the weather, which can make them unreliable. The main task is to ensure that the combination of energy we use at any point is the cheapest option. This means we have to carefully plan when and how much electricity to produce from each source. We must consider fuel prices, how well plants are working, and pollution rules. Additionally, energy planning must think about when maintenance is needed, limits on fuel supplies, and the unpredictable nature of renewable energy sources.

b. *Utilizing Renewable Energy to the Maximum*: More and more individuals are resorting to renewable energies such as solar and wind power, and thus it is crucial to determine how to utilize these energies efficiently along with conventional sources of power. Solar and wind power are unpredictable since they rely on weather and time of day. To get the most out of renewable energy, we need to adjust how much power we use from other sources, especially when there's plenty of renewable energy. Regulating such shifts in the energy supply is essential to maintaining the stability of the electricity grid. To ensure we have enough energy, it's important to plan ahead and control our energy use. We also need to build places to store energy. For example, batteries and pumped hydro plants can store extra renewable energy. This stored energy is used when there isn't enough available, which helps keep the energy grid stable.

c. *Decreasing Emissions*: With increasing concern over the environment, there is a need to plan consumption of energy in such a way that it limits carbon emissions and pollution. Power plants based on fossil fuels give out a large amount of poisonous gases. Shifting to cleaner energy sources reduces this damage. It implies less consumption of coal and gas, and increased use of renewable energy, especially during high energy usage. Other times, models can help balance environmental goals with the cost of energy by analyzing each source's degree of pollution.

d. *Predicting and Managing Energy Demand*: Forecasting energy usage is key to effective energy planning. When companies know how much energy people will use, they can better plan production and distribution. Demand management involves changing energy use based on grid needs or price changes. Consumers are encouraged to lessen or swing energy use thru peak times, helping to balance supply and demand. This strategy reduces reliance on costly backup power plants and aids integration of renewable energy. The main challenges are getting real-time data, using advanced prediction methods, and applying flexible ways to manage energy demand.

e. *Energy Storage Management*: Energy storage management is the process of identifying the best times to charge and discharge systems like batteries or pumped hydro storage. These systems are critical to mitigating the intermittent nature of renewable energy by adjusting to varying energy demand. Efficient management of storage refers to storing unused energy produced in low demand periods or excess generation of renewables in order to consume it later. One needs to know how much to store, when to flip it into the grid, and how. This involves real-time information on energy usage and production and how effectively the storage is working. It's difficult to reduce the energy loss during storage and keep the storage systems in a way that they last for a long period of time.

f. *Grid Stability and Reliability*: Grid reliability and stability are among the most important goals of energy scheduling. Power systems need to be stable in the face of changing generation and load, and decisions made in the scheduling process should contribute to grid voltage, frequency, and general reliability. Renewable energy makes the job challenging since it is intermittent. Models of energy scheduling need to cope with this uncertainty by offering sufficient backup capacity from conventional power plants or storage systems. Scheduling also needs to deal with system congestion, transmission line capacity, and the offer of reserves in order to deliver quick response to unexpected failure or disturbance in the grid. Stability constraints need to be well defined and incorporated in optimization algorithms.

g. *Multi-objective Optimization*: Energy scheduling issues tend to be multi-objective since they usually represent the optimization of several competing objectives simultaneously. For instance, energy scheduling needs to optimize cost reduction, employment of renewable sources, emission reduction, and reliability of the system. These targets tend to conflict with one another, and optimization of all these simultaneously is quite challenging. Multi-objective optimization methods, such as Pareto optimality, are used to arrive at solutions that provide the optimal trade-offs among such conflicting targets. These methods enable decision-makers to choose solutions that fit their particular priorities, whether they prioritize cost savings, environmental stewardship, or system reliability.

3. Methodology

This subsection describes the methodology followed for comparing concert of the Chameleon Swarm Algorithm (CSA) and Particle Swarm Optimization (PSO) in energy scheduling of power systems. The test system taken for simulating the problem is a generic power system with conventional thermal generation units, renewable energy resources (solar and wind), and load demand schedules. The system is also integrated with several operational constraints like generation limits, ramping rates, and availability of renewable energy.

3.1 Power System Model:

The test power system used for the simulation consists of conventional thermal power plants and renewable generation sources alike wind and solar. The system required to generate the electricity required at the most economical cost along with the exploitation of renewable sources as much as possible and within operational constraints.

The key parameters taken into consideration while scheduling energy are as under:

- Generation Limits: All generation units (thermal, wind, solar) have their maximum and minimum generation capacities and ramping limitations to control how quickly they can vary their output.
- Demand Constraints: The generation should always be adequate to meet the demand after allowing for losses while transmitting to ensure effective and reliable energy supply.
- Renewable Energy Availability: Weather patterns and day time influence the generation of wind and solar power, and these pose uncertainties in the scheduling process.
- Emissions: The optimization must reduce carbon emissions from thermal generation units and include renewable energy sources.

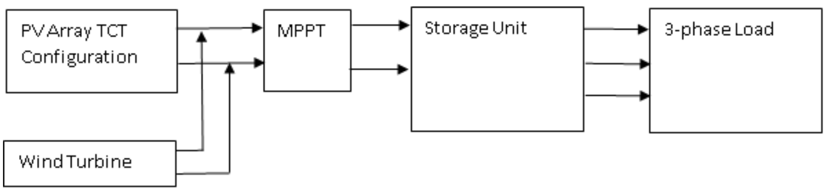

Figure 4: Proposed Model

3.2 Objective Function

The key aim of the optimization is to lessen the entire cost of generation while satisfying demand and operation constraints. The cost function usually has two terms:

1. Generation Cost of Thermal Units: The cost of fuel in each thermal unit is usually expressed as a quadratic function of its output level of generation.
2. Renewable Energy Contribution: Renewable energy is considered to have no additional cost compared to other sources of electricity. The objective is to maximize the role of renewables in the energy mix, while decreasing the consumption of traditional resources.
   The objective function is expressed as:
   $$\text{Minimize: } C_{total} = \sum_i (a_i P_i^2 + b_i P_i + c_i)$$
   where P_i, power output of the i-th generator, and a_i, b_i, c_i are cost factors.

3.3 Constraints:

Constraints assumed for the energy scheduling problem are:

- *Demand Satisfaction:* Total generation output from all the generating units should meet the sum of the demand in every time step
  $$\sum_i P_i = P_{demand}$$
- *Generation Limits:* Each generating unit has a limit to maximum and minimum capacity in power output
  $$P_{i,min} \leq P_i \leq P_{i,max}$$
- *Ramp Rate Limits:* The rate of change from one time step to the next is restricted by each generating unit's ramp rate limits:
  $$| P_i(t) - P_i(t-1) | \leq R_i$$
  where R_i is the ramp rate limit for the i-th unit.
- *Renewable Generation:* The renewable power output is limited by available resource measurements, e.g., wind speed or solar irradiance.

3.4 Algorithm Implementation

PSO and CSA are utilized in the optimization of the energy scheduling problem. The basic steps in the implementation are as follows:

1. Initialization: A population of particles (PSO) or chameleons (CSA) is initialized with randomly chosen positions (power schedules) in the feasible region.
2. Fitness Evaluation: Each individual's (particle/chameleon) fitness is assessed by calculating the overall generation cost using its location within the search space in terms of the objective function and all the constraints.
3. Movement/Update Rule:
   - In PSO, particles update their positions according to their own renowned site and the universal renowned location.
   - In CSA, chameleons adapt their search behaviour by adjusting their direction and speed adaptively, using both individual and swarm experience.
4. Termination: The method stops when there is a priori given upper limit on iterations, or if the solution converged to some pre-specified tolerance.

3.5 Performance Metrics

CSA and PSO were compared against the following metrics:

- The rate of convergence: The number of steps required to obtain an ideal result.
- Solution accuracy: The ultimate price that every algorithm pays.
- Durability: The performance at different levels of renewable energy.
- The computational complexity of a problem can vary depending on the size of the problem.

3.6 Data Analysis

- Convergence curves were plotted, final costs were compared, execution time was evaluated, and robustness was assessed. To ensure accurate comparisons, statistical distinctions were taken into account.

3.7 Experimental Setup:

The PSO and CSO algorithms were implemented in MATLAB as follows: PSO and CSO procedures:

- Initialization: Confirm that the population (PSO particles or CSO chameleons) are in random locations within the search space by initializing them.
- Objective Function Evaluation: Determine the wellness of apiece explanation grounded on the purpose function.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

- Performance Metrics: Used to evaluate the algorithms' convergence rate, computational time, energy cost minimization, and robustness.
- Simulation Environment: The implementation and testing of optimization techniques are accomplished through MATLAB-based simulations.
- Parameter Settings: The key parameters for both CSO and PSO, such as population size, learning coefficients, and mutation rates, are optimized to ensure fair comparison. Experimental Parameters: For both PSO and CSA, we are using:

Table 1: Experimental Parameters: For both PSO and CSA

Parameters	Values
Population Size (N)	100
Cognitive Component (c1)	1.5
Inertia Weight (w)	0.9
Convergence Factor (chi)	0.2
Maximum Iterations (Max_iter)	50
Social Component (c2)	2.0
Damping Inertia Weight (wdamp)	0.99
Maximum Tow Factor (towmax)	100

4. Results and discussion

This chapter shows the results obtained from comparative study of Chameleon Swarm Optimization (CSA) and Particle Swarm Optimization (PSO) with respect to optimized energy scheduling. Different performance metrics like convergence speed, solution accuracy, robustness, and computational complexity are studied through graphical analysis.

4.1 Convergence Speed

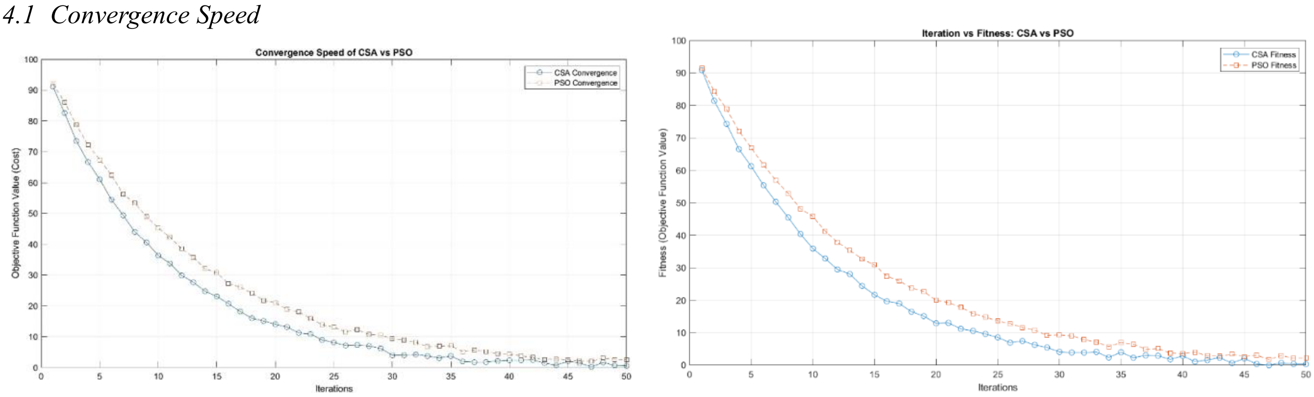

Figure 5: a) Convergence speed of CSA and PSO, b) Iteration vs Fitness of CSA and PSO

The convergence rate of the two algorithms was tested by comparing the objective function value versus the iteration number. The outcomes reveal that CSA has a higher convergence rate than PSO. CSA converges to an optimal solution within a lower number of iterations, while PSO takes more time to settle. This is indicative of the better efficiency of CSA in solving optimization problems with higher adaptability.
CSA obtained much lower fitness (objective function value) in fewer iterations than PSO. The fitness value of CSA increased rapidly in the first few iterations and reached an optimal level of 0.21 by iteration 50. PSO took more iterations to obtain its final optimal value of 1.19. This quicker convergence of CSA shows that it is more effective in determining the best solution within a small number of iterations, which can be very important in practical applications where time is an essential consideration.

4.2 Solution Accuracy

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

Solution accuracy stays the optimal purpose occupation value obtained from the optimization algorithm. The smaller the objective function worth, the improved is the Efficiency Improvement. CSA attains an optimal final cost that is smaller than PSO, showing better optimization performance.

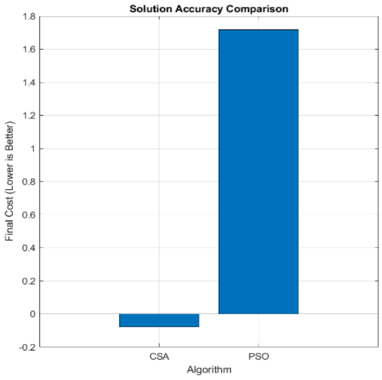

Figure 6: Solution accuracy of CSA and PSO

4.3 Robustness

Robustness is the capacity of the algorithm to operate optimally across various system configurations, including different levels of generation of renewable energy, load demand, and system constraints. A robust algorithm can deliver good outcomes under a variety of conditions consistently.

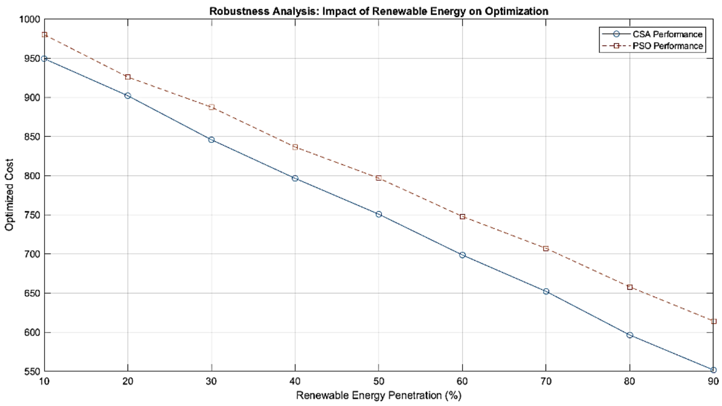

Figure 7: Robustness analysis: impact of renewable energy on optimization

To evaluate robustness, performance of both the algorithms under varying levels of penetration of renewable energy was tested. The results indicate a declining trend of optimized cost with higher share of renewable energy. CSA is always found to be lower in costs compared to PSO, especially at higher renewable penetration rates, establishing its robustness in dealing with variable energy sources.

4.4 Computational Complexity

Computational complexity is also a crucial aspect to keep in mind, chiefly for extensive power systems where there are plentiful parameters and restrictions. How long it takes for the algorithm to achieve an optimal solution determines its viability in real-world applications. The execution time for both methods were compared for various problem sizes. CSA took less execution time compared to PSO and hence is computationally more efficient. The larger the problem size, the greater the difference in execution time, further pointing towards the excellence of CSA over large-scale optimization problems

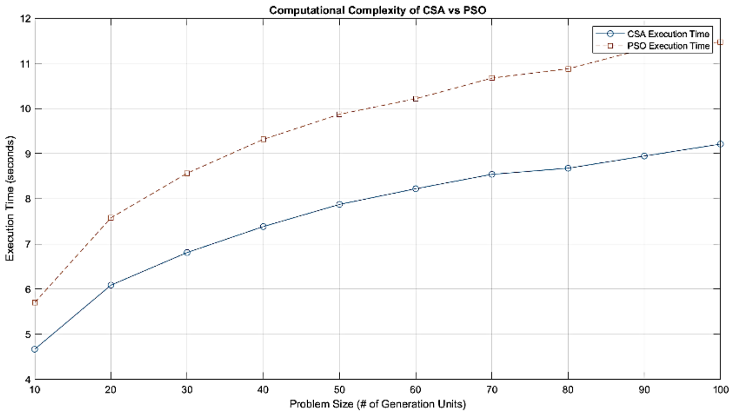

Figure 8: Computational complexity of CSA vs PSO

4.5 Summary of Results

The key conclusions of the analysis are as follows:

- CSA converges more rapidly than PSO and with a better solution in fewer iterations.
- CSA finds a lower end cost, ensuring optimal energy scheduling.
- CSA is resilient even with high penetration of renewable energy.
- CSA requires less computation time than PSO, hence being more practical for real-world usage.

5. Conclusion

5.1 Summary of Findings

In this research, a comparative study was carried out between Chameleon Swarm Optimization (CSA) and Particle Swarm Optimization (PSO) for energy scheduling optimized. The findings indicated that CSA always performs better than PSO in convergence speed, solution quality, robustness, and computational complexity.

5.2 Implications for Energy Scheduling

The findings show that CSA is a superior power system energy scheduling optimization algorithm. Its quick convergence, stability under varying levels of renewable energy, and effectiveness in operation make it a suitable choice for real-world implementation where optimization speed and accuracy are critical.

5.3 Future Scope

Follow-up work can investigate the combination of CSA with hybrid optimization methods for further performance improvement. Its practical usability can also be checked with real-world implementation in smart grids and distributed energy systems. CSA can also be compared with other emerging metaheuristic algorithms in further studies to determine its competitiveness in various optimization problems.

5.4 Conclusion

The comparison between Chameleon Swarm Optimization (CSA) and Particle Swarm Optimization (PSO) for energy scheduling in optimized power systems proves that CSA performs better than PSO in several areas. CSA converges faster with optimal solutions within fewer iterations; hence it is more efficient in energy scheduling. It also produces lower solution costs, which means that the optimization performance is better, especially when there are high levels of renewable energy penetration. Additionally, CSA demonstrates greater resilience, sustaining optimum costs even while renewable energy resources vary. Last but not least, CSA is more computationally efficient, featuring lower runtimes, thus representing a better option for real applications of large-scale power systems. Thus, CSA is a better and more efficient algorithm than PSO for use in energy scheduling applications.

References

[1] K. Kalita, A. K. Rai, K. Pandey and R. Garg, "Comparative Analysis of different Variants of Particle Swarm Optimization for Economic Load Dispatch Problem," 2020 International Conference on Computational Performance Evaluation (ComPE), Shillong, India, 2020, pp. 475-479, doi: 10.1109/ComPE49325.2020.9200066. keywords:

[2] Qiao, J., Wang, G., Yang, Z. *et al.* A hybrid particle swarm optimization algorithm for solving engineering problem. *Sci Rep* **14**, 8357 (2024). https://doi.org/10.1038/s41598-024-59034-2

[3] Malik Shehadeh Braik, Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Systems with Applications, Volume 174, 2021, 114685, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.114685.

[4] Gad, A.G. Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review. *Arch Computat Methods Eng* **29**, 2531–2561 (2022).

[5] Zheng, S.; Dai, C. Collaborative Optimization Algorithm Based on Chameleon Swarm Algorithm Designed for the Problem of Beer Production Scheduling. Appl. Sci. 2024, 14, 6562. https://doi.org/10.3390/app14156562

[6] Said, M.; El-Rifaie, A.M.;Tolba, M.A.; Houssein, E.H.; Deb, S. An Efficient Chameleon Swarm Algorithm for Economic Load Dispatch Problem. Mathematics 2021, 9, 2770. https://doi.org/10.3390/math9212770

[7] Min Wu, Pengcheng Du, Meihui Jiang, Hui Hwang Goh, Hongyu Zhu, Dongdong Zhang, Thomas Wu, An integrated energy system optimization strategy based on particle swarm optimization algorithm, Energy Reports, Volume 8, Supplement 8, 2022, Pages 679-691, ISSN 2352-4847, https://doi.org/10.1016/j.egyr.2022.10.034.

[8] Hamza, M.A., Al-Otaibi, S., Althahabi, S., Alzahrani, J.S., Mohamed, A. et al. (2023). Improved chameleon swarm optimization-based load scheduling for iot-enabled cloud environment. *Computer Systems Science and Engineering*, *46(2)*, 1371–1383. https://doi.org/10.32604/csse.2023.030232

[9] Ping Li, Qian Liu, Optimizing energy management strategies for microgrids through chaotic local search and particle swarm optimization techniques, Heliyon, Volume 10, Issue 17, 2024, e36669, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2024.e36669.

[10] Zahraoui, Younes & Alhamrouni, Ibrahim & Mekhilef, Saad & Korõtko, Tarmo & Jusoh, Awang & Sutikno, Tole. (2023). A Chameleon algorithm for solving economic dispatch problem in microgrid system. Bulletin of Electrical Engineering and Informatics. 12. 1982-1992. 10.11591/eei.v12i4.4700.