International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 2, Mar-Apr-2026, ISSN (Online): 2395-566X

A Review-Based Comparative Study of Metaheuristic Techniques for Optimal Power Flow Optimization

Madhusudan Kumar[1], Abhinav Kumar Singh[1]
[1]Dept. of Electrical and Computer Engineering, Bharati Vidyapeeth (Deemed to be University) College of Engineering, Pune, India
Guide: Prof. Vishal Mehtre

Abstract— Optimal Power Flow (OPF) is a crucial problem in power systems that involves generating power at minimum cost while ensuring safe and feasible operation. This problem is normally solved by using mathematical approaches but they are not always effective due to the problem's complexity. In this context, researchers have started to apply smart, nature-inspired algorithms such as Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). These techniques help find better solutions, even for complex problems. In this paper, we analyse these algorithms in terms of speed, accuracy, computational effort and robustness. After analysing results of various research papers, we find what algorithm works best under various power system conditions.

Keywords—Optimal Power Flow, Metaheuristic Algorithms, Particle Swarm Optimization, Genetic Algorithm, Economic Dispatch, Nonlinear Optimization, Power System Operation, Convergence Characteristics.

I. INTRODUCTION

The power system is getting more complicated as electricity demand continues to grow and more renewable energy is integrated [1]. For these systems to run effectively, it is necessary to determine the amount of power to be generated by each generator in order to minimize the total cost while meeting all technical constraints. This is referred to as Optimal Power Flow (OPF) [2]. OPF has historically been approached using mathematical optimisation methods like gradient-based or linear programming. These techniques perform well for small-scale and well-behaved systems, but may not be suitable for highly nonlinear and non-convex systems, as is the case with power systems. So, they can get trapped at a local (non-optimal) solution or need complicated calculations.

To address these issues, metaheuristic algorithms, which draw inspiration from natural phenomena such as evolution, social behaviour, and foraging, have been introduced [8]. Examples include the Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). These approaches are versatile, don't need gradient information and can search very large spaces efficiently. Introducing new renewable energy sources such as solar and wind makes the OPF problem even more complex because of their variable nature [3]. In this article, we discuss and compare various metaheuristic methods for solving the OPF problem. The comparison is made through key factors such as the speed of convergence, accuracy, computational complexity and robustness of the algorithms. This helps us to identify the best method in certain situations and gives us insight for future research and potential applications [4].

II. LITERATURE REVIEW

In recent years, many researchers have investigated the application of metaheuristic algorithms in solving challenging optimization problems such as Optimal Power Flow (OPF) [5]. The popularity of these algorithms stems from their ability to solve more complex problems, such as nonlinear, non-convex and large-scale problems, than conventional mathematical methods.

The first experiments in metaheuristic optimization demonstrate that techniques like Genetic Algorithm (GA), Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) can be used to solve single and multi-objective optimization problems in different engineering applications [6]. These algorithms mimic natural phenomena and are able to effectively search large solution spaces. PSO has been one of the most popular metaheuristic techniques because of its simplicity and quick convergence. According to survey papers, PSO has been successfully applied to numerous complex engineering problems, including power system optimization, as it has a lower number of parameters and fast convergence. Similarly, GA has also been widely applied to OPF problems because of its excellent global search capability and avoidance of local minima.

Comparative studies of GA, PSO and ACO have been carried out in several cases [7]. These studies indicate that although GA offers superior solutions, it may take longer. Comparatively, PSO has a faster convergence with lesser computational load. ACO, while widely used in discrete optimization problems, is not as popular in continuous problems such as OPF. Recent review literature points out that metaheuristic algorithms are well adapted to modern power systems due to their versatility [8]. They are well

suited for dealing with multi-dimensional and complex engineering problems. But it has also been pointed out by researchers that this approach can be slow in some cases, and may get stuck in local optima [9]. Moreover, research indicates that hybrid algorithms, which integrate two or more algorithms (such as GA-PSO), can enhance the algorithm's performance by combining exploration and exploitation [10]. Such approaches are emerging as a critical area of research in OPF.

In conclusion, the literature shows that metaheuristic algorithms are an efficient and feasible method of solving OPF problems. But no one method performs universally better, and selection of a method depends on the type of problem, system size and accuracy requirements.

III. METAHEURISTIC TECHNIQUES

A. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is one of the most well known and popular metaheuristic algorithms in engineering and computer science [11]. It was first introduced in 1995 by Dr. Eberhart and Dr. Kennedy and completely avoids the traditional calculus-based gradient methods, making it ideal for non-linear and non-convex problems (such as Optimal Power Flow).

1).Biological Inspiration

The PSO algorithm is based on the collective behaviour of biological swarms, such as a flock of birds looking for prey and a school of fish escaping predators [12]. The particles are the potential solutions and update their direction based on their best experience and the swarm's best experience.

2).The Tuning Parameters

The efficiency of the PSO is highly dependent on controlling three parameters in the velocity equation [13]:

• Inertia Weight (w): controls the momentum. A large value (e.g., 0.9) will make the particle fly in a straight line, and thus explore the whole search space. A lower value (e.g., 0.4) makes the particle slow down, favouring local exploitation to find a local optimum. It is usual to gradually reduce w from 0.9 to 0.4 during the iterations.

• Cognitive Coefficient (c_1): controls the influence of the particle's own memory (Pbest).

• Social Coefficient (c_2): controls the particle's trust in the swarm (Gbest).

3).Master Equations of PSO

$$V_i^{t=1} = wv_i^t + c_1r_1(P_{best},I - x_i^t) + c_2r_2(G_{best} - x_i^t) \quad (1)$$

$$X_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

where Vi is the velocity, Xi is the position, Pbest is the personal best, Gbest is the global best, and r_1, $r_2 \in [0,1]$ are random numbers.

MATLAB Pseudocode

```
if particle(i).Cost < particle(i).Best.Cost
   particle(i).Best.Position = particle(i).Position;
   particle(i).Best.Cost     = particle(i).Cost;
   if particle(i).Best.Cost < GlobalBest.Cost
      GlobalBest = particle(i).Best;
   end
end
```

Fig. 1. MATLAB Algorithm Of PSO.

This code checks whether the new location the particle has found is better than its previous best. If so, it also updates its "Personal Best" and, if this is the best spot found by the swarm, it becomes the "Global Best" which all the other particles follow.

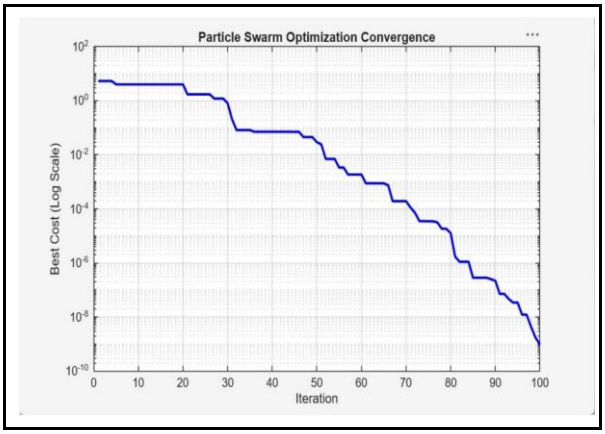

Fig. 2. Convergence graph of PSO (cost vs iteration).

The above figure depicts a very successful optimization. The swarm successfully converged on the optimal solution in 100 iterations, decreasing the cost function from about 10 (at the beginning) to an extremely small number close to 10^{-9}. Thus, the algorithm has successfully found the optimal (or close to optimal) solution.

B. Cuckoo Search (CS) Algorithm

The Cuckoo Search (CS) Algorithm is one of the most intriguing and effective metaheuristic algorithms [14]. Whereas Particle Swarm Optimization (PSO) is based on cooperation, Cuckoo Search is based on parasitism, Darwinian selection and random walks. Due to the difficulty of Optimal Power Flow (OPF) - with many local minimums due to generator valve-point effects and physical constraints on the network - Cuckoo Search is well suited to tackle this problem.

1).Biological Inspiration

CSA is inspired by the behaviour of some cuckoo species which lay eggs in the nests of host birds [15]. Eggs that go unnoticed (good solutions) hatch and the eggs are kept, but eggs that are detected (poor solutions) are thrown out and replaced.

2).The Master Movement Equation

The Master Movement Equation to determine a cuckoo's new position is:

$$X_i^{t+1} = x_i^t + \alpha \square \text{Levy}(\lambda) \qquad (3)$$

where $X_i(t+1)$ is the new position of cuckoo i; $X_i(t)$ is the current position; α is the step size scale, in OPF usually calculated as $\alpha = \alpha_0 \cdot (X_i(t) - G_{best})$; ⊕ means entry-wise multiplication; Lévy(λ) is the random step generated from the Lévy distribution.

3).The "Heavy-Tailed" Math of Lévy Flights

Lévy flights produce a series of small steps occasionally interrupted by very long jumps, allowing the algorithm to escape local minima [16]. The step length s follows a heavy-tailed distribution:

$$\text{Le'vy} \quad \sim s^{-\alpha} \ (1 < \alpha <= 3) \qquad (4)$$

4).Discovery and Replacement

A fraction p_a of the worst nests are abandoned and rebuilt at random positions, modeled as [6]:

$$S = u/|v|^{1/\beta} \qquad (5)$$

MATLAB Pseudocode of CSA

```
for i = 1:nNests
   newPosition = nest(i).Position + alpha * LevyFlight();
   newPosition = max(newPosition, VarMin);
   newPosition = min(newPosition, VarMax);
   newCost     = CostFunction(newPosition);
   if newCost < nest(i).Cost
      nest(i).Position = newPosition;
      nest(i).Cost     = newCost;
   end
   if nest(i).Cost < GlobalBest.Cost
      GlobalBest = nest(i);
   end
end
```

Fig. 3. MATLAB Algorithm Of CSA.

The code uses an odd mathematical jump (Lévy flight) to randomly reallocate the cuckoo from a valid (within grid boundaries) place. Then, it evaluates the fitness of the new place and if it is better (i.e. cheaper) than the previous one, it accepts it, thus possibly improving the record of the whole group.

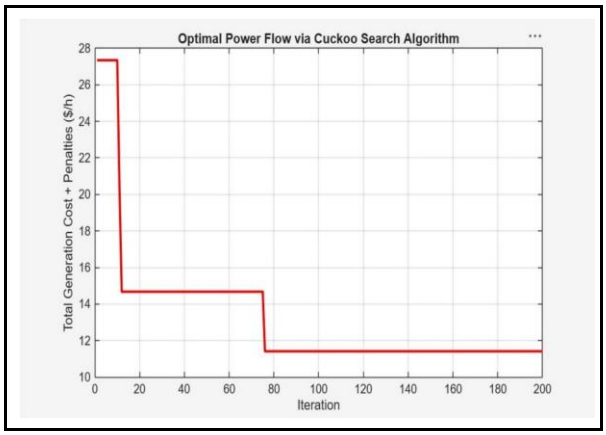

Fig. 4. Convergence graph of CSA (cost vs iteration).

The CSA convergence curve demonstrates solid global exploration, and the ability to jump out of local minima (some Lévy jumps) before fine-tuning to a better optimum [17].

C. Grey Wolf Optimizer (GWO)

Another powerful metaheuristic algorithm is the Grey Wolf Optimizer (GWO) [18]. Whereas PSO is a flock of birds and Cuckoo Search makes teleports, GWO is a team, with a strict hierarchy, and surrounding tactics. Because the OPF problem is highly complicated, with a great risk for getting trapped by "false" optimal solutions, the collective decision-making of GWO is very advantageous.

1) The Biological Inspiration: The Pack Hierarchy

GWO mimics the social hierarchy of grey wolves with: Alpha (α) as the leader, Beta (β) as the advisor, Delta (δ) as the supporter and Omega (ω) as the obedient. The first three wolves decide which direction the pack should move towards the prey [19].

2) Encircling the Prey

Wolves encircle the prey using:

$$D = |C.X_p(t) - X(t) \qquad (6)$$

$$X(t+1) = X_p(t) - A.D \qquad (7)$$

- **D** : The physical distance between the wolf and the prey.
- $\mathbf{X_p}$: The position of the prey (in GWO, the "prey" is the best solution found so far).
- $\mathbf{X_t}$**:** Position of the wolf in iteration t.
- **A and C :** Coefficient vectors (more on these later).

3) The Hunt (Master Equations)

In the real world, say, Optimal Power Flow, the wolves don't know where the "perfect" prey lies. How do they know how to move.They follow the top three leaders: **Alpha (α), Beta (β)**

and **Delta (δ).** All other wolves (the Omegas) have to find their own distance to each of the leaders and compute the average.

Step A: Calculate the distance to the three leaders:

$$D\alpha = |c_1.x\alpha - x| \quad (8)$$
$$D\beta = |c_2.x\beta - x| \quad (9)$$
$$D\delta = |c_2.x\delta - x| \quad (10)$$

Step B: Calculate the step direction toward each leader:

$$X_1 = x\alpha - A_1.D\alpha \quad (11)$$
$$X_2 = x\beta - A_2.D\beta \quad (12)$$
$$X_3 = x\delta - A_1.D\delta \quad (13)$$

Step C: Average them together for the final move:

$$X(t+1) = (x_1 + x_2 + x_3)/3 \quad (14)$$

Why GWO is Excellent for Optimal Power Flow

- It doesn't get trapped like PSO: in PSO, if the Global Best particle falls into a bad local minimum, the entire swarm falls into it as well. In GWO, the Omegas know three different leaders (Alpha, Beta, Delta). If the Alpha is stuck in a local minimum, the Beta and Delta are not, and move the pack away from the crash.
- Smooth exploration-to-exploitation transition driven by the linearly decreasing parameter a [20].

The full GWO main loop integrates the encircling and hunting equations into an iterative search procedure [21].

Main Loop of GWO

```
for it = 1:MaxIt
   for i = 1:nWolves
      A1 = 2*a*rand - a;  C1 = 2*rand;
      D_alpha = abs(C1*Alpha.Pos - wolf(i).Pos);
      X1 = Alpha.Pos - A1*D_alpha;
      % similarly compute X2, X3 from Beta, Delta
      wolf(i).Pos = (X1 + X2 + X3)/3;
   end
   a = 2 - it*(2/MaxIt);
end
```

Fig. 5. MATLAB Algorithm Of GWO.

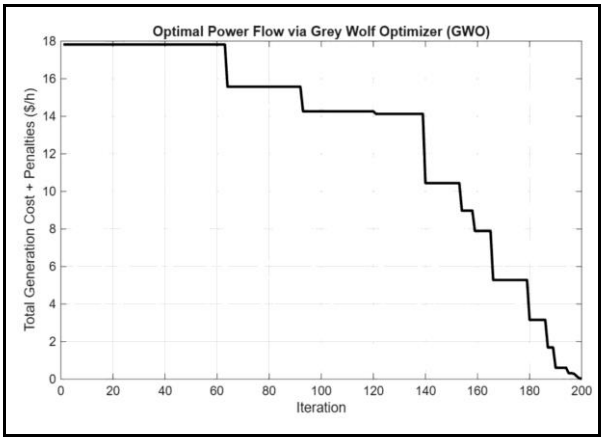

Fig. 6. Convergence graph of GWO (cost vs iteration).

IV. COMPARATIVE ANALYSIS

Benchmarking Particle Swarm Optimization (PSO), the Cuckoo Search Algorithm (CSA) and the Grey Wolf Optimizer (GWO) demonstrates unique modes of operation, with benefits and drawbacks for computer-aided power system analysis [22].

- Particle Swarm Optimization (PSO): PSO is well known for its ease of implementation and rapid convergence [23]. It employs a balance of memory and intelligence of the swarm to quickly probe the search space. But its early convergence habit of following a single "Global Best" leaves it open to getting trapped in local minimums, often when applied to highly constrained real world power systems.
- Cuckoo Search Algorithm (CSA): CSA is an excellent global search algorithm, primarily because it incorporates the heavy-tailed Lévy flights. In mathematical terms this is like "teleporting" the algorithm, allowing it to strongly escape the local minima that normally trap PSO. It is the most easily tuned algorithm (in terms of the number of parameters) but the step-size calculations are more complex, making it slightly slower, but it virtually assures a highly optimal final grid design.
- Grey Wolf Optimizer (GWO): GWO offers the most reliable and stable optimization algorithm of the three. Through an established leadership structure (Alpha, Beta and Delta), the algorithm avoids the problem of search agents being misled by an apparent optimum. The timer parameter naturally steers the pack from global exploration at the beginning of the run to local exploitation at the end.

Final Verdict

PSO is still a good option for fast calculations of simple grid models. If the grid has enormous complexities and numerous local minimums, CSA's random jumping is the most effective way to find the "best" optimal region. But for overall stability, reliability and smooth convergence in typical OPF problems, GWO is the best algorithm, striking a balance between exploration and exploitation to squeeze maximum cost reduction while meeting grid constraints.

V. REFERENCES

[1] S. Duman, J. Li, L. Wu, and U. Guvenc, "Optimal power flow with stochastic wind power and FACTS devices: a modified hybrid PSOGSA with chaotic maps approach," Neural Computing and Applications, vol. 32, no. 12, pp. 8463–8492, 2020, pp. 8463–8470.

[2] M. A. Taher, S. Kamel, F. Jurado, and M. Ebeed, "An improved moth-flame optimization algorithm for solving optimal power flow problem," International Transactions on Electrical Energy Systems, vol. 30, no. 3, e12243, 2020, pp. 1–25.
[3] A. M. Shaheen, R. A. El-Sehiemy, and S. M. Farrag, "A novel adequate bi-level reactive power planning strategy," International Journal of Electrical Power & Energy Systems, vol. 119, 105807, 2020, pp. 1–14.
[4] H. Buch and I. N. Trivedi, "On the efficiency of metaheuristics for solving the optimal power flow," Neural Computing and Applications, vol. 32, no. 23, pp. 17241–17251, 2020, pp. 17242–17249.
[5] M. H. Sulaiman and Z. Mustaffa, "Optimal power flow incorporating stochastic wind and solar generation by metaheuristic optimizers," Microsystem Technologies, vol. 27, no. 9, pp. 3263–3277, 2021, pp. 3265–3270.
[6] M. Abdo, S. Kamel, M. Ebeed, J. Yu, and F. Jurado, "Solving non-smooth optimal power flow problems using a developed grey wolf optimizer," Energies, vol. 11, no. 7, 1692, 2020, pp. 1–18.
[7] K. Nuaekaew, P. Artrit, N. Pholdee, and S. Bureerat, "Optimal reactive power dispatch problem using a two-archive multi-objective grey wolf optimizer," Expert Systems with Applications, vol. 168, 114334, 2021, pp. 1–13.
[8] A. A. El-Fergany and H. M. Hasanien, "Salp swarm optimizer to solve optimal power flow comprising voltage stability analysis," Neural Computing and Applications, vol. 32, no. 9, pp. 5267–5283, 2020, pp. 5269–5278.
[9] S. S. Reddy, "Optimal power flow with renewable energy resources including storage," Electrical Engineering, vol. 102, no. 2, pp. 685–695, 2020, pp. 686–693.
[10] R. Ben Salem, M. Ben Salah, and A. Chebbi, "Optimal power flow solution using a hybrid PSO–GWO algorithm with renewable integration," Energy Reports, vol. 7, pp. 4422–4434, 2021, pp. 4424–4432.
[11] B. Mahdad, "Optimal reconfiguration and reactive power planning based fractal search algorithm: a case study of the Algerian distribution electrical system," Engineering Science and Technology, an International Journal, vol. 23, no. 1, pp. 230–243, 2020, pp. 232–240.
[12] S. Khunkitti, A. Siritaratiwat, and S. Premrudeepreechacharn, "Multi-objective optimal power flow problems based on Slime Mould Algorithm," Sustainability, vol. 13, no. 13, 7448, 2021, pp. 1–24.
[13] H. Pulluri, R. Naresh, and V. Sharma, "An enhanced self-adaptive differential evolution based solution methodology for multi-objective optimal power flow," Applied Soft Computing, vol. 91, 106195, 2020, pp. 1–17.
[14] M. Ebeed, A. Alhejji, S. Kamel, and F. Jurado, "Solving the optimal reactive power dispatch using marine predators algorithm considering uncertainty in load and wind-solar generation," Sustainability, vol. 12, no. 22, 9434, 2020, pp. 1–20.
[15] A. M. Shaheen, A. R. Ginidi, R. A. El-Sehiemy, and S. S. M. Ghoneim, "A forensic-based investigation algorithm for parameter extraction of solar cell models," IEEE Access, vol. 9, pp. 1–20, 2021, pp. 4–15.
[16] S. R. Salkuti, "Optimal power flow using multi-objective glowworm swarm optimization with renewable resources," International Journal of Green Energy, vol. 18, no. 14, pp. 1547–1561, 2021, pp. 1549–1558.
[17] M. Z. Islam, N. I. A. Wahab, V. Veerasamy, H. Hizam, N. F. Mailah, J. M. Guerrero, and M. N. Mohd Nasir, "A Harris Hawks optimization based single- and multi-objective optimal power flow considering environmental emission," Sustainability, vol. 12, no. 13, 5248, 2020, pp. 1–25.
[18] A. Khelifi, B. Bentouati, and S. Chettih, "An efficient chaotic flower pollination algorithm for optimal power flow with FACTS devices," Electrical Engineering, vol. 103, no. 2, pp. 957–979, 2021, pp. 960–972.
[19] H. Buch, I. N. Trivedi, and P. Jangir, "Moth flame optimization to solve optimal power flow with non-parametric statistical evaluation validation," Cogent Engineering, vol. 7, no. 1, 1832806, 2020, pp. 1–22.
[20] M. Riaz, A. Hanif, S. J. Hussain, M. I. Memon, M. U. Ali, and A. Zafar, "An optimization-based strategy for solving optimal power flow problems in a power system integrated with stochastic solar and wind power energy," Applied Sciences, vol. 11, no. 15, 6883, 2021, pp. 1–22.
[21] A. M. Shaheen, R. A. El-Sehiemy, A. Alharthi, S. S. M. Ghoneim, and A. R. Ginidi, "Multi-objective jellyfish search optimizer for efficient power system operation based on multi-dimensional OPF framework," Energy, vol. 237, 121478, 2021, pp. 1–18.
[22] K. Lenin, "Real power loss reduction by Duponchelia fovealis optimization and enriched squirrel search optimization algorithms," Soft Computing, vol. 24, no. 23, pp. 17863–17873, 2020, pp. 17865–17871.
[23] D. Saha, A. Datta, and P. Das, "Optimal coordination of directional overcurrent relays in power systems using symbiotic organism search optimisation technique," IET Generation, Transmission & Distribution, vol. 14, no. 6, pp. 1077–1085, 2020, pp. 1079–1083.