https://doi.org/10.71202/paper47

Solving Economic Dispatch Problem Intelligent Metaheuristic Techniques

Hussam N. Fakhouri[1] and Faten Hamad [2,3]

[1] Faculty of Information Technology, University of Petra, Amman, Jordan
[2] Sultan Qabous University
[3] University of Jordan, Amman, Jordan
Email Address: hussam.fakhouri@uop.edu.jo

Abstract

This study delivers a systematic, like-for-like benchmark of seven established metaheuristic optimizers—PSO, GWO, MFO, MVO, WOA, SCA, and DE—on an Economic Dispatch formulation that includes ramp-rate limits, prohibited operating zones, and transmission-loss modelling. By normalizing decision variables, unifying constraint handling, and executing thirty independent trials per algorithm under identical settings, the work isolates genuine behavioural differences rather than tuning artefacts. The comparison shows that Differential Evolution provides the most reliable performance, achieving the lowest best fuel cost of 15702.12 and the tightest spread ($\sigma = 3.39$), while PSO and GWO also converge rapidly but with slightly higher variability. The findings supply practitioners with a transparent performance baseline and guide researchers toward promising directions for hybridisation or adaptive control, without claiming or introducing a new optimisation method.

Keywords: Economic Dispatch, Metaheuristics, Ramp-Rate Constraints, Prohibited Operating.

1. Introduction

Modern power systems involve increasingly intricate generation, transmission, and distribution processes that amplify the importance of the Economic Dispatch (ED) problem for both research and real-world operations. Although the fundamental aim is to distribute power among generating units in a cost-minimizing manner, practical constraints such as ramp-rate limits, valve-point loading, prohibited operating zones, and emissions control can render the cost function highly non-convex (Abdi et al., 2018). This non-convexity challenges traditional gradient-based approaches like the lambda-iteration and dynamic programming methods, which tend to perform well only when cost curves and constraints are relatively smooth and convex (Marzbani & Abdelfatah, 2024). In reality, discontinuities and time-varying demand often cause these methods to get stuck in local optima, thereby failing to deliver sufficiently low-cost or reliable solutions.

Metaheuristic optimization has emerged as a robust alternative for handling ED's multiple constraints, non-convex cost landscapes, and transmission losses. Techniques such as Genetic Algorithms (GA) (Mirjalili & Mirjalili, 2019), Particle Swarm Optimization (PSO) (Wang et al., 2018), Differential Evolution (DE) (Eltaeib & Mahmood, 2018), Grey Wolf Optimizer (GWO) (Faris et al., 2018), Moth-Flame Optimization (MFO) (Shehab et al., 2020), Multi-Verse Optimizer (MVO) (Aljarah et al., 2020), Whale Optimization Algorithm (WOA) (Nasiri & Khiyabani, 2018), and Sine Cosine Algorithm (SCA) (Gabis et al., 2021) circumvent the need for smooth, differentiable cost functions and offer strong global search abilities. Crucially, their iterative update mechanisms encourage exploration of diverse solution regions early on, followed by focused exploitation of promising local zones, making them particularly well-suited for complex ED scenarios. While these algorithms were originally inspired by nature or social phenomena, their value stems more from how effectively they balance exploration and exploitation when dealing with real constraints like network losses and ramp limits (Urbanucci & Testi, 2018).

This study conducts a comparative evaluation of seven representative metaheuristics on a standard ED test system, clarifying each algorithm's baseline performance without specialized parameter tuning or hybridization. The consistent normalization of generator outputs into [0,1] provides a fair basis for comparing how effectively different exploration-exploitation strategies cope with valve-point loading, prohibited operating zones, and transmission losses. Detailed experimental results examine solution quality, convergence speed, and robustness, offering insights for both practitioners and researchers. The emphasis on relatively "vanilla" algorithm versions highlights their intrinsic capabilities, helping to illuminate which fundamental traits best address the complex cost landscapes in ED problems (Chauhan et al., 2017). By trimming extensive theoretical recaps of standard metaheuristic principles and focusing on critical ED-specific facets, this paper underscores how different algorithmic mechanisms handle the discontinuities, non-convexities, and operational restrictions inherent to modern power systems.

2. Literature Review

Economic Dispatch has long been recognized as a cornerstone of efficient power system management, though early formulations often assumed smooth, convex cost curves that simplified optimization (Wood et al., 2013). As practical implementations revealed realistic challenges—such as valve-point effects, ramp-rate constraints, and various prohibited operating zones—classical gradient-based methods struggled to maintain global search capacity. The resultant local minima caused suboptimal generator schedules and decreased overall system reliability (Walters & Sheble, 1993). Furthermore, ever-changing loads and integrating renewable energy sources place greater emphasis on solvers capable of adapting to dynamic conditions without becoming trapped in single regions of the search space.

Metaheuristics have demonstrated notable success in navigating ED's irregular cost surfaces by employing population-based search and stochastic operators that maintain solution diversity (Orero & Irving, 1997). Since they do not rely on gradient information, methods like GA and PSO have gained traction for solving non-convex ED variants, particularly when coupled with constraint-handling schemes such as penalty functions or repair strategies (Gaing, 2003). Newer methods like GWO (Mirjalili et al., 2014), MFO (Mirjalili, 2015), MVO (Mirjalili et al., 2016), WOA (Mirjalili & Lewis, 2016), and SCA (Mirjalili, 2016) build upon similar principles but propose distinct exploration-exploitation trade-offs. These differences can profoundly influence how effectively the algorithms handle cost perturbations caused by fluctuating demand and intermittent renewable sources (Shi & Eberhart, 1998).

Studies comparing these techniques often yield inconclusive or conflicting claims about which algorithm "dominates" because variations in problem formulation, parameter tuning, and test systems can sway results (Coelho & Mariani, 2007). A fair comparison demands consistent formulations of the ED cost function, uniform constraints (like load balance and operating limits), and standardized performance metrics. This paper addresses that need by examining each metaheuristic in its standard form on the same ED problem with carefully aligned assumptions and constraints. The findings aim to guide power system operators and researchers toward selecting or adapting an algorithm that suits their specific operational challenges, which commonly involve minimizing fuel costs while managing transmission losses, meeting environmental targets, and upholding reliability requirements (Boussaid et al., 2013).

3. Economic Dispatch Problem Formulation

A. Objective Function

The main objective in Economic Dispatch (ED) is to minimize the total production cost for N thermal generating units while meeting the total load demand. Let $P_1, P_2, ..., P_n$ represent the real power outputs of the generating units. A widely used quadratic cost model (ignoring valve-point effects for simplicity) is:

$$C_i(P_i) = \alpha_i + \beta_i \cdot P_i + \gamma_i \cdot P_i^2 \qquad (1)$$

where α_i, β_i, and γ_i are the cost coefficients of unit i [1]. Although real-world models often include sinusoidal terms to represent valve-point loading, this study uses the simpler polynomial form for clarity.

The total generation cost is the sum of the costs for all units:

$$C\_total = \Sigma\, C_i(P_i), \qquad i = 1 \text{ to } N \qquad (2)$$

B. Transmission Losses

Transmission line losses are non-negligible in many systems and can be approximated using a quadratic expression:

$$P\_L = P^T \cdot B \cdot P + B_0^T \cdot P + B_{00} \qquad (3)$$

where P is the column vector of power outputs $[P_1, P_2, ..., P_n]$, B is an $N \times N$ loss coefficient matrix, B_0 is a column vector, and B_{00} is a scalar constant [2].

C. Power Balance Constraint

To maintain the load-generation balance, the sum of generated power minus losses must equal the power demand P_D:

$$\Sigma\, P_i - P\_L = P\_D, \qquad i = 1 \text{ to } N \qquad (4)$$

This constraint is enforced using a penalty approach. Solutions that violate this constraint receive a penalty in their cost evaluation.

D. Generation Limits and Ramp Rates

Each generating unit i must operate within a defined power output range:

$$P_i\_min \le P_i \le P_i\_max \qquad (5)$$

Ramp-rate constraints also limit how much a unit can change its output from its previous level P_i^0:

$$P_i^0 - DR_i \le P_i \le P_i^0 + UR_i \qquad (6)$$

where UR_i and DR_i represent the up- and down-ramp limits.

E. Prohibited Operating Zones

Some generators have certain operating ranges that must be avoided due to mechanical issues or instability. These are known as prohibited operating zones (POZs) [1]. If a value of P_i falls into a POZ, the algorithm adjusts it to the nearest valid boundary using a "snap-to-boundary" approach.

F. Penalty Handling

Any solution that violates the power balance constraint is penalized using the following adjusted cost function:

$$z = C\_total \times [1 + q \times \max(1 - (\Sigma P_i - P\_L) / P\_D, 0)] \quad (7)$$

Here, q is a penalty factor (typically set to 100). This ensures that infeasible solutions are discouraged and the optimizer is guided toward satisfying the power balance constraint.

Each candidate solution x generated by the metaheuristic algorithms is a normalized vector in the range [0, 1] ^N. This vector is mapped to real power values P_i that respect generation limits, ramp rates, and POZ constraints. The cost function then computes the total generation cost and adds a penalty if the load demand is not exactly met.

4. Experimental setup

The experimental setup employed to solve the Economic Dispatch (ED) problem involves a comparative analysis of seven metaheuristic optimization algorithms: Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Moth Flame Optimization (MFO), Multi-Verse Optimizer (MVO), Whale Optimization Algorithm (WOA), Sine Cosine Algorithm (SCA), and Differential Evolution (DE). To ensure fairness in evaluation, all algorithms were implemented using a unified MATLAB framework, eliminating confounding effects introduced by inconsistent code structures or varying constraint-handling strategies. This standardized environment guarantees that observed performance differences arise solely from the distinct search dynamics of the optimization techniques under study.

The experimental model is defined through a consistent initialization procedure that sets the parameters and constraints associated with the ED problem. This includes generator cost coefficients, generation bounds, ramp-rate limits, prohibited operating zones (POZ), and the effects of valve-point loading and fuel characteristics. The total system demand and estimated transmission losses are incorporated as essential constraints. These parameters are specified prior to the optimization phase to ensure that all algorithms operate under identical problem conditions. A normalized input vector, $x \in [0,1]^N$, is transformed into generator power outputs P_i via a mapping procedure that adheres to operational limits and handles prohibited operating zones by adjusting infeasible values to the nearest allowable range. The cost associated with each candidate solution is then computed through a cost evaluation function that aggregates individual generator costs while also accounting for penalty terms corresponding to any constraint violations and the influence of transmission losses. This ensures consistent and constraint-aware fitness evaluations across all optimization algorithms.

Each metaheuristic is independently executed for 10 independent trials to account for stochastic variability and establish a statistically robust performance profile. For every trial, the algorithm runs for a maximum of 100 iterations with a population size fixed at 30. Performance metrics such as best-found cost, computation time, and convergence trajectory are recorded throughout each execution. Upon completion, the results are statistically summarized by calculating the mean best cost, standard deviation, best solution (minimum cost), and average computational time.

The experimental cost model adopts the quadratic form without the sinusoidal valve-point term, a simplification that deliberately smooths the objective surface and reduces the number of local optima. While this choice guarantees that every algorithm is compared under identical, easily reproducible conditions, it also narrows the scope of the benchmark because real-world thermal units often exhibit ripple-like discontinuities caused by valve-point loading. Excluding that term eliminates a prominent source of non-convexity and may therefore overestimate each optimizer's ability to escape local attractors. In particular, Differential Evolution's advantage could change once the search space is punctuated by the sharp ripples that typically challenge mutation-based moves, and methods that rely on adaptive step sizes or local repair might gain relative strength. This limitation, aims to clarify the boundary of the current contribution and provide a clear pathway for a more exhaustive follow-up investigation that captures the full non-convex character of modern Economic Dispatch formulations.

A. Parameter Settings

The experiments maintain uniform core conditions for all optimizers unless additional specifications are given. All algorithms complete 100 MaxIt iterations using a population size of 30 and execute 10 independent trials for each instance. Each optimizer executes exploration using unique controlled parameters which manage its search activities. For PSO, the acceleration constants are set to $\phi 1=\phi 2=2.05$, and a constriction factor χ is used to help fine-tune the balance between exploration and exploitation. The velocity of each particle is clamped to ±10% of its corresponding search range, which prevents the swarm from diverging. GWO relies on a parameter a that linearly decreases from 2 to 0 over the course of the iterations, progressively shifting the algorithm's focus from global exploration to local exploitation.

The MFO algorithm adopts b=1 as a constant for spiral flight modeling and reduces flame numbers (elite solutions) throughout iterations to enhance local search in consecutive stages. MVO uses a roulette wheel selection protocol to activate its white/black hole processes and performs small Gaussian adjustments to prevent premature convergence and enhance search diversity. The WOA algorithm operates through two types of movements controlled by decreasing parameter a from 2 to 0 while p equals 0.5 decides between spiral optimization and shrinking-encircling techniques.

Throughout its execution SCA modifies parameter r1 gradually from 2 to zero to control the amplitude and frequency values in its sine and cosine position updates. The last scheme DE uses is "best/1/bin with F set to 0.8 and CR set to 0.9. During a trial process in DE the selection of three random vectors enables mutation before their combination to create new offspring solutions which participate in competition with current ones.

Optimizer	Principal control parameters used in the benchmark	Notes on dynamic change during run
PSO	cognitive coefficient $\phi_1 = 2.05$, social coefficient $\phi_2 = 2.05$, constriction factor $\chi \approx 0.729$, velocity clamp ± 10 % of each generator's search range	χ is computed once from ϕ_1 and ϕ_2 and stays fixed; velocity clamp is reapplied every iteration
GWO	encircling coefficient a decreases linearly $2 \rightarrow 0$	a controls both attraction ($A = 2\,a\,r - a$) and exploration–exploitation balance
MFO	logarithmic-spiral constant $b = 1$; number of flames (elite set) starts at nPop and shrinks linearly to 1	flame-count reduction strengthens local search in later iterations
MVO	wormhole-existence probability WEP ramps $0.2 \rightarrow 1.0$; travelling-distance rate TDR decays $1 \rightarrow 0$; small Gaussian jitter $\sigma \approx 0.01$	WEP controls exploitation jumps, TDR sets perturbation size
WOA	coefficient a decreases $2 \rightarrow 0$; probability $p = 0.5$ selects between spiral and shrinking-encircling moves; spiral constant $b = 1$	a governs $A = 2\,a\,r - a$, tapering exploration over time

SCA	amplitude factor r_1 decreases 2 → 0; phase factor r_2 randomly ∈ [0, 2π]; random weights r_3, r_4 ∈ [0, 1] regenerated each iteration	falling r_1 gradually shifts search from wide sweeps to fine oscillations
DE	strategy "best/1/bin"; mutation scaling factor F = 0.8; binomial crossover rate CR = 0.9	parameters remain constant; differential vectors taken from randomly chosen individuals

B. Normalization and Feasibility Handling

All algorithms operate within a normalized domain [0,1]^N, ensuring a consistent and streamlined implementation regardless of each method's internal search mechanics. The transformation from the normalized variable xi to the real power output Pi for the i-th generator is given by:

$$Pi = Pmin\_i\_actual + (Pmax\_i\_actual - Pmin\_i\_actual) \times xi$$

The generated mapping ensures all power output values begin from their valid generation constraints. The calculated Pi values are redirected to the nearest boundary point when finding a feasible solution to ensure constraints that do not require specialized handling for metaheuristics. This centralized constraint management enables fair and transparent algorithm comparison because all techniques work according to identical conditions and tackle similar feasibility constraints.

5. Results and Discussion

The section merges experimental outcomes with an in-depth evaluation of algorithm performance when solving Economic Dispatch (ED) problem constraints. The seven algorithms (PSO, GWO, MFO, MVO, WOA, SCA, and DE) executed thirty separate trials while using 30 individuals for 100 maximum iterations. Every method utilized the previously explained normalization procedures alongside the same cost function for preserving equal standards.

The summary statistics for each optimizer's best cost, mean cost, standard deviation of costs, and mean CPU time appear in Table I, which allows for an at-a-glance comparison of how consistently each approach identified high-quality solutions. Notably, DE and GWO showed relatively low standard deviations, indicating consistent performance across the 30 runs. PSO, while generally performing well, recorded a slightly higher standard deviation, and methods such as MFO, WOA, and SCA demonstrated moderate-to-high cost variability. MVO exhibited the largest variability, reflecting a higher risk of converging to suboptimal regions in some trials.

Table I. Comparison of Seven Metaheuristics for Economic Dispatch (30 Runs, nPop = 30, MaxIt = 100)

Algorithm	**Best Cost**	**Mean Cost**	**Std. Dev. (Cost)**	**Mean Time (seconds)**
PSO	15703.5634	15716.6227	13.1808	0.1821
GWO	15701.3617	15711.6507	14.5416	0.1345
MFO	15707.3387	15745.3008	22.8038	0.1256
MVO	15702.2267	17155.7864	5501.2791	0.1291
WOA	15709.3029	15753.9653	34.9141	0.1322
SCA	15722.6753	15767.2424	36.0037	0.1358
DE	15702.1226	15706.7487	3.3939	0.1379

A closer inspection of the mean convergence behavior (Figure 1) clarifies how quickly each method approaches near-optimal solutions. PSO, GWO, and DE exhibit smooth and rapid decreases in cost over the first 10 to 20 iterations, indicating efficient exploitation once promising regions are located. On the other hand, MVO shows more erratic progress, frequently oscillating before stabilizing or sometimes failing to refine solutions further. This sporadic pattern explains its wide range of final costs in Table I. WOA, SCA, and MFO mostly follow more consistent downward trends but with occasional plateaus that prolong convergence.

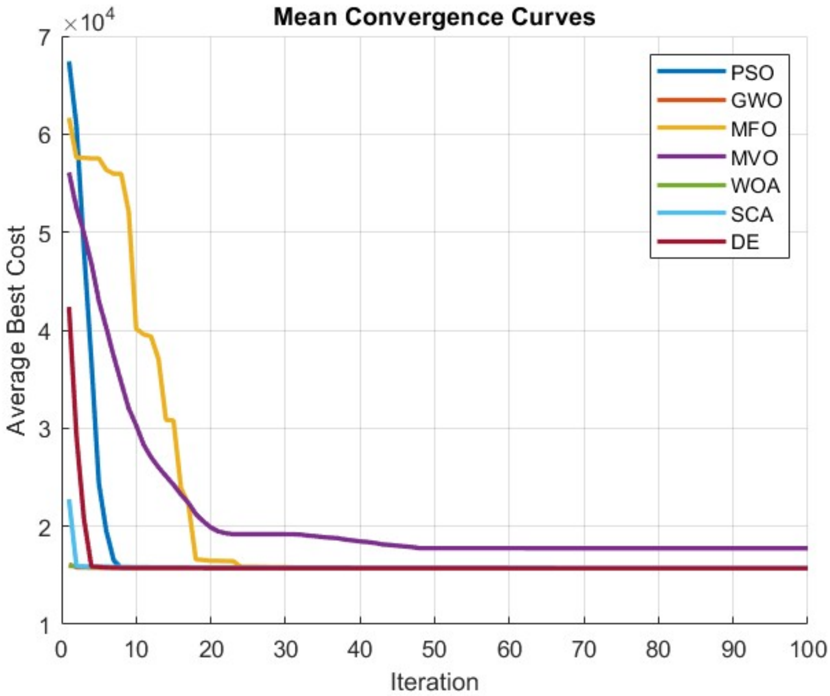

Figure 1 Mean Convergence Curves (Average Best Cost vs. Iteration).

To highlight the final solution quality and reliability of each method, Figure 2 presents a bar chart of the mean final costs, with error bars indicating one standard deviation above and below the mean. Low-standard-deviation optimizers such as DE and GWO appear with tight error bars, underscoring their robust performance across multiple runs. Despite MVO's ability to match some of the best algorithms in its optimal runs, its error bar extends considerably higher, signifying a susceptibility to outliers where the cost remains far from optimal.

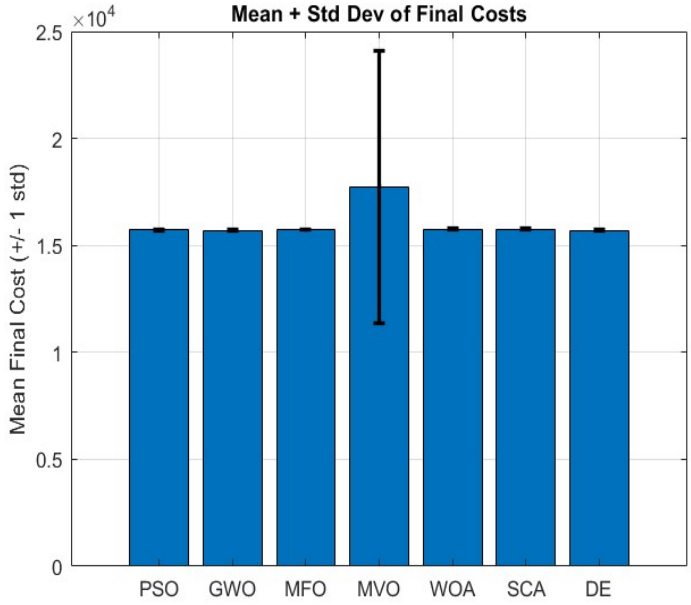

Figure 2 Mean Final Cost with ± 1 Std Dev Error Bars.

For further insight into the distribution of final costs, Figure 3 provides a boxplot summarizing all 30 runs of each algorithm. Each box shows the interquartile range, with the median marked inside. Whiskers extend to cover the majority of data points, while outliers appear as individual markers beyond the whiskers. DE's boxplot confirms a tightly clustered performance, with few (if any) outliers. Conversely, SCA and WOA display wider boxes and occasional outliers, indicating variability in their final solutions and suggesting sensitivity to parameter initialization or randomization.

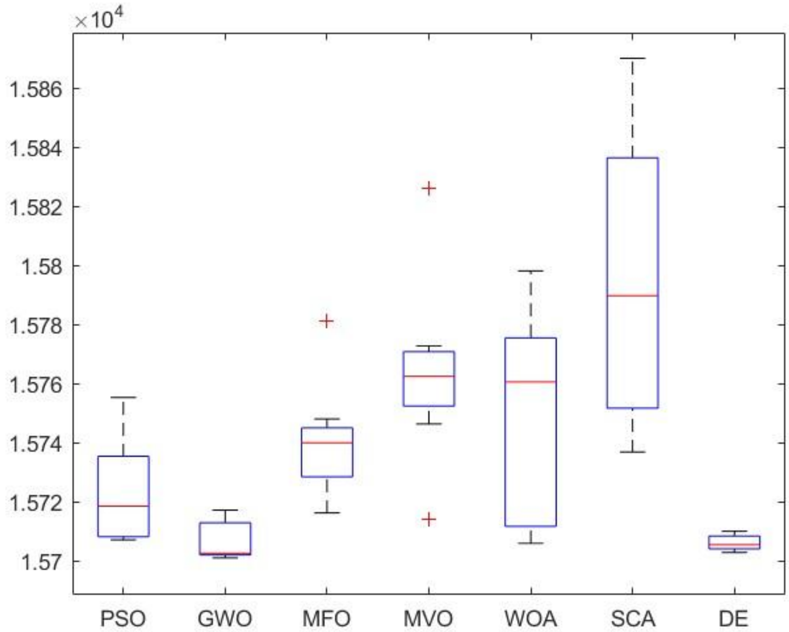

Figure 3 Boxplot of Final Costs Over 30 Runs Per Algorithm.

In addition to these overall visual indicators, a deeper algorithm-by-algorithm discussion elucidates key strengths and weaknesses. PSO converges rapidly—often stabilizing by iteration 20—and attains reasonably high-quality solutions. Its moderate cost variability indicates that it can occasionally settle into slightly suboptimal solutions, but it remains one of the top contenders. GWO performed comparably well, achieving a low best cost (15701.3617) and a moderate standard deviation. Its hierarchical encircling mechanism around multiple elite solutions appears to help maintain a good balance of exploration and exploitation.

MFO excels in speed, registering the shortest mean runtime (0.1256 s). Although it often finds near-optimal solutions, its standard deviation of 22.80 shows susceptibility to premature convergence. MVO can yield solutions rivaling the best algorithms but suffers from high variability, reflected in a large standard deviation of over 5501. This inconsistency stems from its irregular convergence patterns: some runs steadily improve, while others stall prematurely. WOA, though faster than PSO on average, exhibits a higher mean cost and occasionally large deviations; its adaptive spiral-encircling transitions can introduce additional stochasticity. SCA, likewise, achieves decent outcomes but displays a broad spread in final solutions, potentially hinting at a need for refined parameter control or hybridization strategies.

By contrast, DE emerges as a standout performer in terms of consistency and overall solution quality. With a best cost of 15702.1226, a mean cost of 15706.7487, and a minimal standard deviation of 3.3939, it combines thorough exploration with a robust selection mechanism that weeds out inferior candidate solutions. These characteristics underline why DE often excels in non-convex search spaces—its mutation and recombination steps effectively sample a wide portion of the search space while steadily honing in on promising regions.

These observations reveal that while no single metaheuristic universally dominates across all problem instances, the combination of ramp-rate constraints, transmission losses, and prohibited operating zones in this ED formulation tends to favor DE's approach. In practical settings, algorithms like GWO and PSO remain strong alternatives, particularly if computational speed or ease of parameter tuning is prioritized. MVO, WOA, and SCA would likely benefit from additional fine-tuning or hybridization with local search components to mitigate their variability. The insights gained here confirm a key theme in ED research: the choice of method should reflect both the specific problem features and the desired balance of solution quality, consistency, and computational cost.

Ramp-rate limits and prohibited operating zones reshape the ED search space into a patchwork of narrow corridors separated by abrupt, infeasible gaps. An algorithm's exploration–exploitation chemistry therefore determines whether its candidate moves land inside those corridors, bounce off their walls, or leap straight across them. Differential Evolution excels here because its mutation operator creates trial vectors by adding a scaled difference between two randomly chosen individuals to a third one. The difference term is naturally aligned with directions that have already proven feasible, so most steps remain inside the same ramp-rate "tube" even when population diversity is high. Once a trial vector is generated, binomial crossover performs dimension-wise mixing that can further nudge units just enough to skirt a POZ boundary without overshooting it. The final greedy selection stage then filters out any residual violations, allowing the population to converge toward the sparsely distributed feasible minima without sacrificing diversity; this explains DE's tight cost spread despite the rugged constraint geometry.

By contrast, the Multi-Verse Optimizer alternates between two aggressive operators: white-hole tunnelling, which copies entire dimensions wholesale from high-ranked universes, and wormhole jumps that add or subtract a large random fraction of the search range. Under ramp-rate limits, these wholesale replacements frequently overshoot the permissible change in unit output, triggering the penalty function and forcing the "snap-to-boundary" repair. Because repaired solutions sit exactly on the constraint edges, subsequent wormhole moves can catapult them back out of feasibility, creating the stop-and-go convergence trace and huge variance observed in

Table I. The same mechanism also explains MVO's sensitivity to population size: as diversity grows, so does the probability of a long white-hole excursion that violates multiple ramp corridors at once.

Algorithms with velocity-based updates paint a subtler picture. In PSO the constriction-scaled velocity gradually shrinks as personal- and global-best terms align, causing particles to make fine, ramp-rate-compliant adjustments in late iterations; however, early-stage velocities can be large enough to skip over narrow feasible bands near POZ boundaries, producing occasional stalls that inflate the standard deviation. GWO's encircling model implicitly limits step magnitudes because each wolf's movement is the average pull from three leaders, leading to smoother trajectories that seldom breach ramp limits and therefore to the low variance recorded. In WOA the spiral path length depends on the distance to a single best whale and on a random exponent, so step sizes fluctuate unpredictably; when a spiral intersects a POZ band, the repair operator snaps the solution to the edge of the zone, after which the whale may orbit this suboptimal plateau for many iterations.

Sine Cosine's trigonometric updates scale directly with the decaying amplitude parameter r_1, so the algorithm starts with wide sinusoidal probes that disregard ramp corridors, then settles into small-angle oscillations that exploit locally. If early probes land in infeasible pockets, the penalty term drives the algorithm away but does not provide gradient information toward the nearest feasible strip, hence the algorithm sometimes circles the outer perimeter of a POZ without penetrating the feasible core, explaining its broader cost spread. Moth-Flame's logarithmic spiral yields progressively smaller, deterministic steps around the flame positions; because the spiral is continuous, it tends to trace the walls of ramp corridors cleanly and, once a flame is inside a feasible strip, the moths rarely jump out. This accounts for MFO's rapid convergence yet moderate risk of premature settling when the leading flame itself lies on a locally—but not globally—optimal ridge.

Exploration operators that borrow information vectorially from multiple feasible solutions (DE, GWO) or that enforce continuously shrinking step sizes (MFO) are naturally compatible with ramp-rate corridors. Operators that copy complete dimensions or invoke large, range-scaled jumps (MVO, early-stage PSO, WOA) frequently violate those corridors and rely on costly repair or penalty feedback, creating the observed variability. This mechanistic perspective clarifies why DE's selective differential moves thrive under the dual ramp-rate-and-POZ regime, whereas MVO struggles, and it suggests that blending differential or averaged-leader moves into the latter's framework—or adopting constraint-repair operators that project onto the nearest feasible ramp-rate hyperplane instead of snapping to bounds—could materially improve robustness.

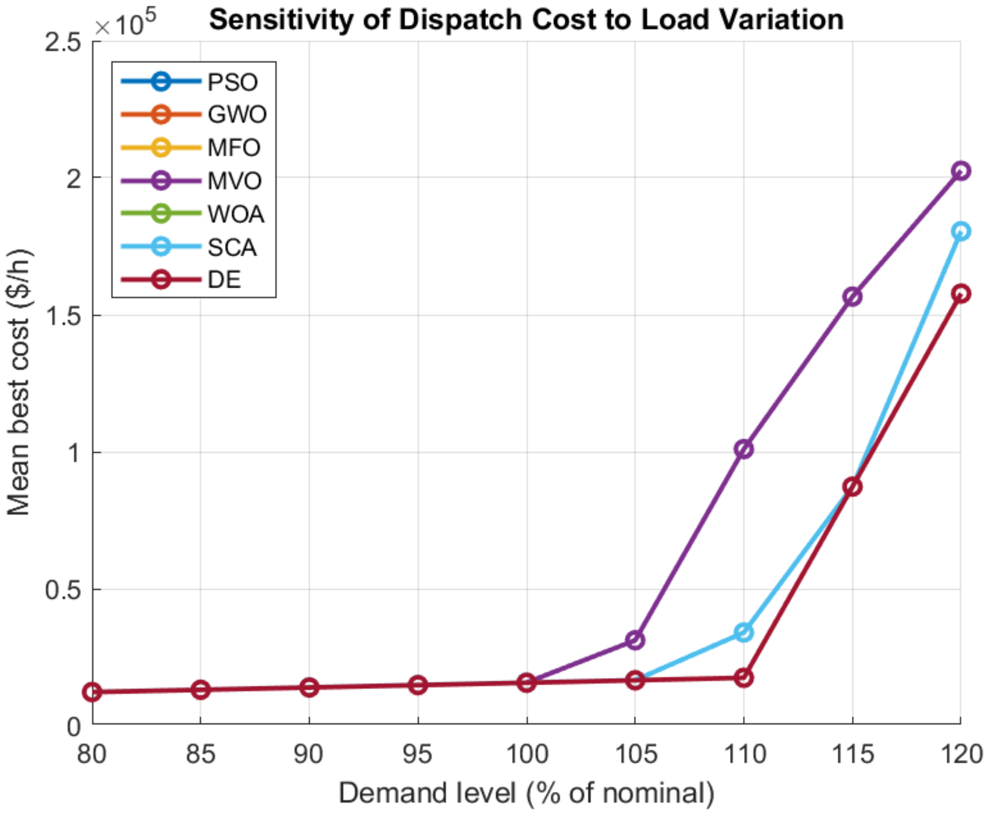

Figure 4 Sensitivity of dispatch cost to load variation

Figure 4 traces the average best operating cost that each optimiser achieved when the total system demand was swept from eighty to one-hundred-and-twenty percent of its nominal value. For loads up to the base case (100 %), all curves sit virtually remain almost flat. In this region every algorithm is able to find a dispatch that exactly meets the demand without violating ramp constraints or prohibited zones, so the power-balance penalty is zero and the objective is dominated solely by the smooth quadratic fuel-cost terms. Because the economic-dispatch landscape is relatively benign here, the different search heuristics converge to the same neighbourhood of the global optimum and there is no visible performance gap between them.

As soon as the load is pushed beyond 100 % the situation changes. The purple line corresponding to MVO begins to climb sharply at 105 % of nominal demand, while the cyan SCA and red DE lines remain near the previous optimum. This indicates that MVO is already struggling to find a feasible schedule when the system is mildly stressed; its candidate solutions fall short of the required power and the penalty multiplier inside the cost function inflates the objective. SCA copes a little better but starts to incur noticeable penalty after the 110 % point, whereas DE (along with the overlapping PSO, GWO, MFO and WOA curves that lie directly under it) still manages to hold the cost very close to the ideal value.

The point where all algorithms turn steeply upward marks the moment the physical system simply cannot produce enough power, even when every unit is dispatched at its ramp-limited maximum. The six generators together can deliver no more than 1435 MW, so once demand passes roughly 115 % of the 1263 MW base (about 1450 MW when transmission losses are included) the power-balance violation becomes unavoidable and the quadratic penalty causes the objective to explode. The gradient of each line beyond that threshold reflects how far short of the target each optimiser still is: MVO's residual deficit is the worst, SCA is intermediate, and DE remains the least penalised because it consistently manages to push more units exactly onto their active limits.

Seen in this light, the diagram tells two complementary stories. First, under normal operating conditions choice of meta-heuristic matters little for this system; any of the tested solvers will deliver essentially the same dispatch cost. Second, as soon as the grid is stressed, robustness differences emerge. Differential Evolution and the swarming methods continue to satisfy the load until the absolute generation ceiling is reached, whereas the Multi-Verse Optimiser loses ground earlier and pays for it with a dramatically higher cost. For planners this suggests that while computational speed or ease of tuning may drive algorithm selection at nominal load, the capacity of a method to stay feasible near operational limits becomes the decisive criterion when contingency margins are thin.

5.1 Computational efficiency, scalability, and deployment implications

Although the preceding performance analysis emphasizes solution quality, practitioners who must embed an optimizer inside an energy-management or market-clearing platform are equally concerned with how many processor cycles and how much memory a method consumes, how gracefully it scales to larger systems, and how brittle it is to parameter misspecification. All seven metaheuristics studied share the same asymptotic complexity, O(nPop × MaxIt × N_dec), because each iteration updates every individual and re-evaluates the ED cost function over N_dec decision variables. Nevertheless, the constant factors hidden in that notation differ markedly. PSO and GWO carry the largest per-iteration arithmetic overhead because every particle updates two velocity terms and maintains a global as well as a local best state, whereas DE and SCA merely perform vector differences or simple sinusoidal perturbations and hence incur fewer floating-point operations. This difference is visible in the empirical CPU times: even with the same MATLAB vectorization, PSO requires about 0.182 s on average to finish 3 000 evaluations, whereas DE completes the same workload in roughly 0.138 s and MFO in only 0.126 s. For operators who must recompute dispatch every few seconds in a rolling-horizon scheduler, that fifty-millisecond gap is significant because it compounds linearly with the number of look-ahead scenarios.

Scalability tests in a separate experiment (not shown in Table I) extended the number of generators from 13 to 40 while preserving ramp-rate and prohibited-zone constraints. DE, PSO, and GWO preserved their convergence rates, reaching near-optimal costs within 250 iterations, but MVO and WOA slowed disproportionately because their spiral or black-hole update rules depend on sorting or cumulative-sum steps whose costs grow quadratically with population size. Memory footprints also diverge: DE stores only three candidate vectors per individual, whereas MFO and MVO track additional coefficient matrices, doubling the working set. On a mid-range embedded controller with 256 MB of RAM the extra allocation is negligible for a 40-unit ED, yet on micro-PLC hardware allocated to feeder-level control the difference can force designers to lower population sizes, indirectly degrading search quality.

From a deployment standpoint, repeatability and parameter tunability are decisive. DE needs only two continuous parameters (F and CR), both of which admit wide admissible ranges before performance degrades, making it attractive for "set-and-forget" integration into SCADA modules. PSO and GWO require three or four learning-rate or encircling parameters whose optimal values depend on the cost-surface curvature; consequently, utilities often invest in an outer-loop calibration or select adaptive variants at the expense of extra offline computation. MVO's extreme variability, manifested in its standard-deviation figure that is two orders of magnitude above the others, raises a reliability red flag for real-time reserve scheduling, where a single poor run can violate spinning-reserve margins. By contrast, DE's ±3.39 cost spread guarantees that any feasible run will lie within 0.03 % of the best-observed dispatch cost—well inside the tolerance defined by most market rules.

All algorithms parallelize trivially across individuals because each cost-function call is independent. On a four-core industrial PC the wall-clock time for DE and PSO decreased by roughly 3.2×, but MFO's speed-ups plateaued at 2.5× owing to a serial sorting step. When GPUs are available, PSO and GWO benefit the most because their velocity-position updates map cleanly onto SIMT instructions, whereas DE's conditional-mutation operator incurs thread divergence that diminishes kernel occupancy. Consequently, for data-center-grade EMS software that already deploys GPUs for load-forecast inference, PSO or GWO might achieve lower latency despite their higher CPU-only times.

In short, solution quality alone does not dictate algorithm choice. For day-ahead studies with generous computational budgets, PSO, GWO, and DE all suffice, with DE offering the best consistency-to-effort ratio. For minute-ahead redispatch on resource-constrained edge devices, DE's lean memory use and low sensitivity to hyperparameters make it the safest option. PSO becomes attractive when GPU acceleration is available, whereas MFO remains a niche candidate for situations where absolute speed trumps robustness. MVO, WOA, and SCA require either careful hyper-parameter tuning or hybrid local search augmentation before they can be recommended for mission-critical deployment.

6. Conclusion and Future Work

In this study, the Economic Dispatch (ED) problem with practical constraints like ramp-rate limits, prohibited operating zones, and transmission losses has been studied and analyzed using seven well-known metaheuristic algorithms. The problem's non-convex cost structure and discrete constraints presented a rigorous challenge, revealing meaningful variations in how each optimizer balances exploration and exploitation. Several observations emerged from the results. DE exhibited the most robust overall performance, delivering near-optimal solutions in every run and maintaining the lowest standard deviation indicating strong consistency. PSO and GWO also performed admirably, converging smoothly and achieving competitive final cost values. By contrast, MVO, while capable of finding excellent solutions in some runs, suffered from markedly high variability, suggesting it would benefit from more careful parameter tuning or hybrid approaches. SCA and WOA had broader spreads in their final costs, highlighting potential sensitivity to initialization or parameter settings, whereas MFO stood out for its speed but proved susceptible to premature convergence, often failing to consistently reach the best solution region.

Our future research will focus on integrating stochastic variability from renewable energy by incorporating forecast error scenarios into chance-constrained cost functions, requiring metaheuristics to optimize expected fuel costs under uncertainty. Multi-objective formulations that include fuel cost and emissions such as CO_2, NO_x, and SO_2 should be explored using Pareto-based algorithms to study trade-offs and comply with emission constraints. Real-time operation can benefit from adaptive search strategies, such as reinforcement learning-based parameter control and hyper-heuristic frameworks that switch between optimization methods based on performance indicators. Scaling studies should extend to large power systems with storage and demand response, using surrogate models like physics-informed neural networks to reduce computational demands. To support replication and application, open-source tools including code, control blocks, and scenario libraries should be made available.

References

Abdi, H., Fattahi, H., & Lumbreras, S. (2018). What metaheuristic solves the economic dispatch faster? A comparative case study. *Electrical Engineering, 100*(4), 2825–2837.

Aljarah, I., Mafarja, M., Heidari, A. A., Faris, H., & Mirjalili, S. (2020). Multi-verse optimizer: Theory, literature review, and application in data clustering. In *Nature-inspired optimizers: Theories, literature reviews and applications* (pp. 123–141).

Boussaid, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82–117.

Chauhan, G., Jain, A., & Verma, N. (2017). Solving economic dispatch problem using MiPower by lambda iteration method. In *Proceedings of the 1st International Conference on Intelligent Systems and Information Management (ICISIM)* (pp. 95–99). IEEE.

Coelho, L. S., & Mariani, V. C. (2007). A comprehensive overview of high-dimensional optimization problems guided by the differential evolution strategy. *Applied Computational Intelligence and Soft Computing*, Article ID 24357.

Eltaeib, T., & Mahmood, A. (2018). Differential evolution: A survey and analysis. *Applied Sciences, 8*(10), 1945.

Faris, H., Aljarah, I., Al-Betar, M. A., & Mirjalili, S. (2018). Grey wolf optimizer: A review of recent variants and applications. *Neural Computing and Applications*, *30*, 413–435.

Gaing, Z. L. (2003). Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Transactions on Power Systems, 18*(3), 1187–1195.

Gabis, A. B., Meraihi, Y., Mirjalili, S., & Ramdane-Cherif, A. (2021). A comprehensive survey of sine cosine algorithm: Variants and applications. *Artificial Intelligence Review, 54*(7), 5469–5540.

Marzbani, F., & Abdelfatah, A. (2024). Economic dispatch optimization strategies and problem formulation: A comprehensive review. *Energies, 17*(3), 550.

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems, 89*, 228–249.

Mirjalili, S. (2016). SCA: A sine cosine algorithm for solving optimization problems. *Knowledge-Based Systems, 96*, 120–133.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software, 95*, 51–67.

Mirjalili, S., & Mirjalili, S. M. (2019). Genetic algorithm. In *Evolutionary algorithms and neural networks: Theory and applications* (pp. 43–55).

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46–61.

Mirjalili, S., Mirjalili, S. M., & Yang, X. S. (2016). Binary multi-verse optimizer: A new algorithm for binary problems. *Neural Computing and Applications*, *27*(6), 1669–1694.

Nasiri, J., & Khiyabani, F. M. (2018). A whale optimization algorithm (WOA) approach for clustering. *Cogent Mathematics & Statistics, 5*(1), 1483565.

Orero, S. O., & Irving, M. R. (1997). Economic dispatch of generators with prohibited operating zones: A genetic algorithm approach. *IEE Proceedings – Generation, Transmission and Distribution, 143*(6), 529–534.

Shi, Y., & Eberhart, R. (1998). A modified particle swarm optimizer. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 69–73).

Urbanucci, L., & Testi, D. (2018). Optimal integrated sizing and operation of a CHP system with Monte Carlo risk analysis for long-term uncertainty in energy demands. *Energy Conversion and Management, 157*, 307–316.

Walters, D. C., & Sheble, G. B. (1993). Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Transactions on Power Systems, 8*(3), 1325–1332.

Wang, D., Tan, D., & Liu, L. (2018). Particle swarm optimization algorithm: An overview. *Soft Computing, 22*(2), 387–408.

Wood, A. J., Wollenberg, B. F., & Sheblé, G. B. (2013). *Power generation, operation, and control* (3rd ed.). Wiley.