



Learning-Driven Adaptive Genetic Algorithm for Real-World Dynamic Energy Systems

Anurak Choeichum¹ and Chaipayorn Khemapatapan²

ABSTRACT

Learning-enabled optimization in dynamic energy systems demands more than high-quality solutions; it requires sustained adaptation under non-stationary and shock-driven environments. This paper proposes Adaptive Genetic Algorithm with Dynamic Operator Scoring and Operator Elimination (AGA-DO), an evolutionary learning framework that explicitly models operator-level learning to achieve long-term robustness. In contrast to conventional genetic algorithms with static or heuristically adaptive operators, AGA-DO continuously evaluates operator contributions, applies temporal smoothing to stabilize learning dynamics, and eliminates persistently ineffective operators to prevent negative knowledge accumulation. Unlike conventional adaptive genetic algorithms that primarily focus on parameter adaptation and peak optimization performance, AGA-DO emphasizes operator-level learning, learning stability, and recovery capability under dynamic environments. A five-phase validation framework is employed to progressively evaluate effectiveness, learning behavior, robustness, and real-world recovery capability. Beyond instantaneous performance metrics, Cumulative Regret Loss (CRL) is adopted as an observational indicator to quantify the accumulated cost of delayed adaptation and repeated suboptimal decisions over time. Results show that benchmark algorithms accumulate substantially higher regret under repeated environmental shocks, whereas AGA-DO achieves the lowest cumulative regret (CRL = 0.81) and the best overall Friedman rank (1.2). These findings suggest that AGA-DO can serve as an effective learning-driven evolutionary optimizer for real-world dynamic energy systems. The proposed framework emphasizes learning stability and retention over aggressive convergence, aligning with practical requirements of real-world adaptive optimization.

Article information:

Keywords: Adaptive Genetic Algorithm, Operator-Level Learning, Dynamic Operator Elimination, Evolutionary Learning, Dynamic Energy Systems

Article history:

Received: February 10, 2026

Revised: June 12, 2026

Accepted: September 2, 2026

Published: October 17, 2026

(Online)

DOI: 10.37936/ecti-cit.2026204.266140

1. INTRODUCTION

Dynamic and non-stationary optimization problems are increasingly encountered in modern artificial intelligence applications, where objectives, constraints, and environmental conditions evolve over time. Such changes may render previously optimal solutions suboptimal or infeasible, posing challenges to computational intelligence methods that rely on stable evaluation criteria [1–4].

Conventional evolutionary and swarm-based algorithms, including genetic algorithms (GA), particle swarm optimization (PSO), and differential evolution (DE), perform well in static or mildly dynamic en-

vironments but may degrade under non-stationary conditions because of premature convergence, loss of diversity, and delayed response to environmental changes [5–9].

Adaptive evolutionary algorithms have therefore been developed, mainly through parameter-level adaptation such as mutation rates [4–12], crossover probabilities [10–15], and other adaptive control mechanisms [16–21]. Although these approaches improve flexibility, they remain largely reactive and may provide limited long-term adaptation under highly dynamic conditions.

Recent studies have explored operator-level adap-

^{1,2}The authors are with the Department of Computer Engineering, College of Engineering and Technology, Dhurakij Pundit University, Bangkok, Thailand, Email: anurak.choeichum@gmail.com and chaipayorn@dpu.ac.th

²Corresponding author: chaipayorn@dpu.ac.th

tation and credit assignment to allocate search effort toward more effective operators [18, 20, 22, 28]. However, many approaches rely on short-term performance feedback, which may lead to unstable learning under non-stationary environments. Moreover, real-world validation and systematic recovery analysis after abrupt disruptions remain relatively limited [30–36].

To address these limitations, this paper proposes AGA-DO (Adaptive Genetic Algorithm with Dynamic Operator Scoring and Operator Elimination), a learning-driven evolutionary framework combining operator-level scoring, temporal stabilization, and dynamic elimination of persistently ineffective operators. A progressive five-phase validation framework evaluates learning effectiveness, robustness, and recovery capability, culminating in real-world shock and recovery experiments without algorithm-specific parameter retuning.

2. RELATED WORK

2.1 Evolutionary Algorithms in Dynamic Environments

Dynamic optimization has been widely studied in evolutionary computation, where changing objectives and constraints require continuous adaptation [1–3]. Existing approaches often rely on diversity preservation, memory mechanisms, or heuristic assumptions about environmental changes, which may limit their effectiveness in complex real-world scenarios.

2.2 Adaptive Genetic Algorithms

Adaptive genetic algorithms have been proposed to enhance flexibility by adjusting control parameters such as mutation rates, crossover probabilities, and population diversity during evolution [12–16], [26]. While parameter adaptation improves responsiveness, it does not fundamentally alter the underlying search operators. Consequently, the algorithm may continue to apply ineffective operators even after environmental shifts, leading to inefficient exploration and delayed recovery.

2.3 Operator Selection and Credit Assignment

Operator selection mechanisms dynamically allocate search effort based on operator performance [18, 20, 22]. However, many credit-assignment approaches rely on short-term feedback, making learning susceptible to noise and transient fluctuations in non-stationary environments.

2.4 Comparative Metaheuristics

Metaheuristics such as PSO, DE, and NSGA-II have been applied to dynamic optimization problems [6–9], [24], [27]. However, their reliance on

population-level dynamics and fixed update rules can limit adaptability under abrupt environmental changes.

2.5 Limitations of Existing Validation Practices

GA-based methods have been applied across diverse real-world engineering and intelligent-system applications, including power-system reconfiguration, nonlinear modeling, energy optimization, scheduling, and other application-specific problems [23], [25], [29–36].

However, these studies primarily emphasize optimization performance within individual application domains, while systematic evaluation of recovery under repeated non-stationary shocks remains limited. Such recovery behavior is particularly underexplored in long-horizon, real-world time-series environments.

Table 1: Related Work Comparison.

Method	Adaptation Level	Learning Stability	Operator Learning	Real-world Validation
GA/AGA	Parameter	No	No	Limited
PSO	Parameter	No	No	Limited
DE	Parameter	No	No	Limited
NSGA-II	Population	Partial	No	Limited
AGA-DO	Operator	Yes	Yes	Yes

The reviewed studies collectively illustrate the evolution from parameter-level adaptation toward operator-level adaptation. However, existing approaches still lack mechanisms for learning stabilization, operator retention, and long-term learning consistency under repeated environmental changes. These limitations motivate the proposed AGA-DO framework.

2.6 Research Gap and Motivation

Based on the above review, existing adaptive evolutionary algorithms suffer from three key limitations:

1. adaptation is predominantly parameter-level rather than learning-driven,
2. operator-level mechanisms lack stability and long-term refinement, and
3. validation rarely considers real-world shock and recovery behavior.

These gaps motivate the development of AGA-DO, as shown in Table 1, which integrates stable operator-level learning with dynamic elimination, and its evaluation through a comprehensive progressive validation framework.

3. DYNAMIC OPTIMIZATION PROBLEM FORMULATION

This section formulates the class of dynamic optimization problems considered in this study. The formulation is presented in a general form to reflect non-stationary optimization challenges commonly encountered in artificial intelligence applications. A

real-world energy system is subsequently introduced as an instance for experimental validation.

3.1 Dynamic Decision Variables

Let the optimization problem be defined over a discrete time horizon $t = 1, 2, \dots, T$. At each time step t , a vector of decision variables $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]$ represents the system control or configuration to be optimized.

Unlike static optimization, the decision variables in dynamic environments must continuously adapt to time-varying conditions, where previously optimal solutions may lose effectiveness as the environment evolves.

3.2 Time-Varying Objectives

The optimization objective is defined as a time-dependent function:

$$\min f(\mathbf{x}(t), t) \quad (1)$$

where $f(\cdot)$ denotes one or more objective functions whose characteristics may change over time. Such changes may arise from external disturbances, evolving system demands, or fluctuating resource availability.

In this study, a multi-objective formulation is adopted to capture trade-offs commonly observed in real-world decision-making processes. The objective functions are evaluated at each time step and aggregated over the time horizon to reflect long-term performance.

3.3 Dynamic Constraints

Dynamic optimization problems are typically subject to time-varying constraints:

$$\mathbf{g}(\mathbf{x}(t), t) \leq 0, \mathbf{h}(\mathbf{x}(t), t) = 0 \quad (2)$$

where constraint boundaries may shift due to environmental changes or system uncertainties. Constraint violations are handled implicitly through feasibility mechanisms to ensure solution validity throughout the optimization process.

3.4 Non-Stationarity and Environmental Changes

A defining characteristic of the considered problem class is non-stationarity, where the mapping between decision variables and objective outcomes changes over time. This non-stationarity may be gradual or abrupt, resulting in shocks that significantly alter the optimization landscape.

An effective computational intelligence framework must therefore exhibit:

- Adaptability to evolving objective functions,
- Robustness to sudden environmental disruptions,

- Recovery capability after performance degradation.

These requirements motivate the learning-driven operator adaptation mechanisms proposed in this work.

3.5 Energy System as a Dynamic Optimization Instance

To evaluate the proposed framework under realistic conditions, a real-world energy system is employed as a dynamic optimization instance. In this context, decision variables represent generation scheduling decisions, while objectives correspond to cost and environmental impact measures that vary over time.

The energy system exhibits strong non-stationary behavior due to fluctuating demand, renewable generation variability, and externally injected shocks. Importantly, the problem formulation remains consistent with the general dynamic optimization framework described above, allowing the proposed method to be assessed as a general computational intelligence approach rather than a domain-specific solution.

3.6 Objective Functions

Table 2: Variable Definition.

Symbol	Definition	Unit
$P_j(t)$	Power Dispatch of Biomass Unit j at time t	MW
$D_{gross}(t)$	Gross electricity Demand	MW
$P^{PV}(t)$	Available PV Generation	MW
$P^W(t)$	Available Wind Generation	MW
$D_{net}(t)$	Net Demand to be Supplied by Biomass	MW
$P_j^{max}(t)$	Maximum Available Biomass Capacity	MW
$C_j(t)$	Time-varying Generation Cost Coefficient	Cost/MWh
e_j	Emission Coefficient	kgCO ₂ /MWh
Δt	Time Step Duration	Hour
$\chi_j(t)$	Time-dependent Shock intensity factor	
β_j	Cost sensitivity coefficient of biomass unit j to the time-dependent shock intensity	Dimensionless

The problem is formulated as a bi-objective optimization task:

$$f_1 = \sum_{t=1}^T \sum_{j=1}^J P_j(t) C_j(t) \Delta t \quad (3)$$

$$f_2 = \sum_{t=1}^T \sum_{j=1}^J P_j(t) e_j \Delta t \quad (4)$$

Net Demand:

- $D_{net}(t) = \max(D_{gross}(t) - P^{PV}(t) - P^W(t), 0)$

Dynamic Cost:

- $C_j(t) = C_j^{base} \cdot (1 + \beta_j \cdot \chi_j(t))$

Constraints:

- $\sum_{j=1}^J P_j(t) \geq D_{net}(t), \forall t$
- $0 \leq P_j(t) \leq P_j^{max}(t), \forall j, t$

- Constraint handling is embedded within the Pareto-based selection mechanism, avoiding explicit penalty parameters.

4. PROPOSED METHOD: AGA-DO

This section presents AGA-DO (Adaptive Genetic Algorithm with Dynamic Operator Scoring and Operator Elimination), an adaptive evolutionary intelligence framework designed for dynamic and non-stationary optimization problems.

Unlike conventional genetic algorithms that rely on fixed operator usage or parameter-level adaptation, AGA-DO introduces learning mechanisms at the operator level, enabling continuous refinement of search behavior based on observed performance over time.

The overall learning-driven workflow of the proposed AGA-DO is illustrated in **Fig. 1**, highlighting the closed-loop interaction between operator selection, performance evaluation, temporal stabilization, and dynamic operator elimination.

Sudden disruptions create discontinuities that invalidate previously effective search behaviors, highlighting the need for learning mechanisms capable of retaining useful experience while suppressing outdated strategies.

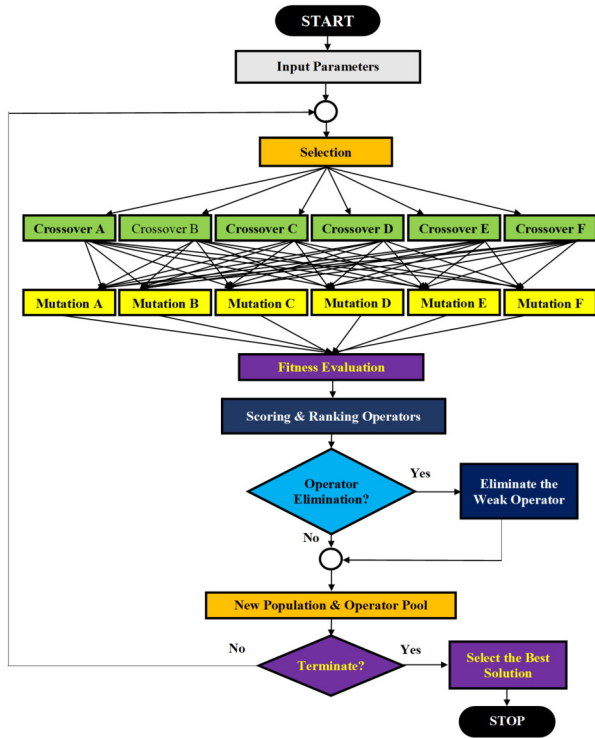


Fig.1: Learning-driven workflow of the proposed AGA-DO framework.

4.1 Operator Pool and Search Diversity

AGA-DO maintains a diverse operator pool consisting of multiple crossover and mutation operators with different exploration-exploitation charac-

teristics. Instead of preselecting a single operator or applying all operators uniformly, the algorithm treats operators as competing strategies whose effectiveness may vary across different stages of the search process and environmental conditions.

This design allows the algorithm to preserve search diversity while providing the flexibility required to respond to changes in the underlying fitness landscape. By decoupling operator usage from static heuristics, AGA-DO creates a foundation for learning-driven adaptation.

4.2 Adaptive Operator Scoring Mechanism

To enable operator-level learning, AGA-DO assigns a performance score to each operator based on its contribution to search progress. After each generation, operators are evaluated according to hypervolume improvement and learning stability.

Rather than relying on instantaneous rewards, the scoring mechanism aggregates performance information over time, allowing the algorithm to distinguish consistently effective operators from those that perform well only under transient conditions.

Operator selection probabilities are updated adaptively based on these scores, guiding the search toward more promising strategies.

Table 3 lists the primitive crossover and mutation operators used to construct the operator pool. In AGA-DO, an executable operator is defined as a composite pair of one crossover operator and one mutation operator.

Table 3: Primitive Crossover and Mutation Operators Used to Construct the Composite Operator Pool.

Category	ID	Operator Type
Crossover	C1	Arithmetic Mean
Crossover	C2	Weighted Arithmetic
Crossover	C3	Uniform Exchange
Crossover	C4	Temporal Segment Swap
Crossover	C5	Differential Crossover
Crossover	C6	Linear Blend
Mutation	M1	Gaussian Perturbation
Mutation	M2	Uniform Perturbation
Mutation	M3	Temporal Flip
Mutation	M4	Random Reset
Mutation	M5	Scaling Mutation
Mutation	M6	Identity (No-op)

Therefore, the active operator pool is not limited to the 12 primitive operators listed in Table 3, but consists of $6 \times 6 = 36$ composite operators:

$$\Omega = \{O_{a,b} = (C_a, M_b) \mid a = 1, \dots, 6, b = 1, \dots, 6\}$$

where C_a denotes the a -th crossover operator and M_b denotes the b -th mutation operator. Thus, the initial operator pool size is $N_{op} = 36$. Operator scoring, selection, and elimination are performed at the

composite-operator level.

4.2.1 Hypervolume-Based Reward Assignment

To evaluate operator effectiveness in a dynamic multi-objective environment, AGA-DO first applies Pareto-based non-dominated sorting to identify the current Pareto-optimal front. The quality of the resulting front is then assessed using the Hypervolume (HV) indicator, which simultaneously reflects convergence quality and solution diversity.

The contribution of each operator is measured according to the improvement in hypervolume obtained after generating offspring solutions. The normalized hypervolume improvement is defined as

$$dHV_i(t) = \frac{HV_{new}(t) - HV_{current}(t)}{\max(|HV_{current}(t)|, \varepsilon)}$$

where $HV_{current}(t)$ and $HV_{new}(t)$ denote the hypervolume values before and after applying operator i , respectively, and ε is a small positive constant used to avoid division by zero.

To improve learning stability and reduce sensitivity to temporary performance fluctuations, a stability term is incorporated into the reward mechanism. The stability penalty is defined as

$$V_i(t) = |dHV_i(t) - dHV_i(t-1)|$$

A larger value of $V_i(t)$ indicates stronger fluctuation in operator contribution and therefore reduces the final reward. The reward assigned to operator i is computed as

$$R_i(t) = \max(R_{min}, 0.35dHV_i(t) - 0.25V_i(t))$$

where $R_{min} = 0.05$ prevents operator starvation and preserves exploration capability.

A minimum reward value of 0.05 is imposed to prevent operator starvation and preserve exploration capability throughout the learning process. The coefficients were determined through preliminary sensitivity analysis conducted during pilot experiments.

Preliminary experiments indicated that these values provided a reasonable balance between responsiveness and learning stability across all validation phases.

The coefficients do not represent normalized weights. They act as scaling factors empirically selected to balance hypervolume improvement and learning stability.

Since hypervolume is computed from the Pareto-optimal front, Pareto dominance information and solution diversity are implicitly incorporated into the reward mechanism through the hypervolume improvement term.

Consequently, operators are rewarded not only for producing better objective trade-offs but also for

maintaining stable performance under changing environmental conditions.

4.2.2 EMA-Based Operator Score Update

To avoid unstable operator selection caused by one-generation reward spikes, AGA-DO updates operator scores using an exponential moving average rather than direct cumulative addition. Let $C_i(t)$ denote the set of offspring generated by operator i at generation t , and let $\bar{R}_i(t)$ denote the mean reward obtained by these offspring. The operator score is updated as

$$\bar{R}_i(t) = \frac{1}{|C_i(t)|} \sum_{c \in C_i(t)} R_c(t)$$

$$S_i(t) = \alpha S_i(t-1) + (1-\alpha)\bar{R}_i(t)$$

where $S_i(t)$ is the EMA-smoothed operator score used for both operator selection and operator elimination. The parameter α controls the balance between accumulated operator experience and the current generation reward. This formulation avoids ambiguity between raw accumulated scores and smoothed scores.

4.2.3 Adaptive Operator Selection Probability

The operator selection probability is determined using a normalized probability distribution that maps operator scores to a probability distribution. This approach ensures that higher-performing operators are more likely to be selected, while still allowing exploration of less frequently used operators.

$$P_i(t) = \frac{A_i(t)(S_i(t) + \varepsilon)}{\sum_{k=1}^{N_{op}} A_k(t)(S_k(t) + \varepsilon)}$$

The operator selection probability (P_i) is computed using the temporally smoothed operator score. This allows operator selection to reflect both recent performance and accumulated learning experience, preventing abrupt probability fluctuations caused by short-term performance variations.

4.3 Dynamic Operator Elimination Strategy

To prevent persistently ineffective operators from consuming computational resources, AGA-DO employs a ranking-based elimination mechanism. Unlike threshold-based approaches, which require manually tuned cut-off values, the proposed strategy evaluates operators according to their relative performance within the active operator pool.

At each evaluation interval, operators are ranked according to their temporally smoothed scores. The operator with the lowest smoothed score is identified as the least effective operator during the recent evaluation interval. To ensure gradual refinement rather than aggressive pruning, only one operator is removed at each elimination event.

The elimination process is executed every 15 generations, allowing sufficient time for operators to accumulate performance evidence before removal decisions are made. Let O_{active} denote the number of active operators at generation t . The operator pool is updated as

$$i^* = \arg \min_{i \in \mathcal{A}(t)} S_i(t)$$

$$A_{i^*}(t+1) = 0, \text{ if } |A(t)| > K_{min}$$

Once eliminated, an operator is permanently removed from the active pool for the remainder of the run.

To preserve search diversity and avoid excessive loss of exploration capability, operator elimination is terminated when the active operator pool reaches a predefined minimum size of 17 operators.

Importantly, operator elimination serves as a long-term learning regulation mechanism rather than an aggressive pruning strategy. The minimum pool size of 17 operators was determined empirically to maintain a balance between operator diversity and computational efficiency based on preliminary sensitivity analysis and experiments that indicated no further improvement in robustness when additional operators were removed.

4.4 Algorithm Workflow and Discussion

As depicted in **Fig. 1**, AGA-DO extends the conventional evolutionary cycle by embedding an operator-level learning loop. Operator performance is continuously evaluated after each generation, accumulated through temporal smoothing, and fed back to regulate future operator selection and elimination decisions. This feedback loop enables the algorithm to retain effective search behaviors while suppressing outdated or ineffective operators under non-stationary conditions.

From a computational perspective, the additional overhead is limited to operator-score updates and probability recalculation, which are relatively small compared with objective-function evaluation in the considered experiments.

By integrating adaptive scoring, temporal stabilization, and dynamic elimination, AGA-DO enables continuous learning at the operator level. This design allows the algorithm to maintain robustness, stability, and recovery capability under changing environments, forming the basis for the progressive validation framework presented in the following section.

Unlike parameter-level adaptive algorithms, the learning loop in **Fig. 1** operates at the operator granularity, enabling long-term behavioral refinement rather than short-term reactive adjustment.

Algorithm 1: AGA-DO Learning Procedure

Input: Initial Population P , Maximum Generation T_{max}
Output: Final Pareto-optimal Solution Set

- 1: Initialize population P
- 2: Initialize active composite operator pool $|A(0)| = 36$
- 3: Initialize operator scores $S_i = 0$
- 4: Initialize operator selection probabilities P
- 5: for $t = 1$ to T_{max} do
- 6: Select composite operator O_i according to $P_i(t)$
- 7: Generate offspring solutions
- 8: Evaluate objective functions
- 9: Construct Pareto front and compute HV
- 10: Compute reward $\bar{R}_i$
- 11: Update operator score $S_i(t)$ using EMA
- 12: Update operator selection probability $P_i(t)$
- 13: if $\text{mod}(t, I_{elim}) = 0$ and $|A(t)| > K_{min}$ then
- 14: $i^* = \arg \min_{i: A_i(t)=1} S_i(t)$
- 15: $A_{i^*}(t+1) = 0$
- 16: else
- 17: $A(t+1) = A(t)$
- 18: end if
- 19: end for
- 20: Return final Pareto front

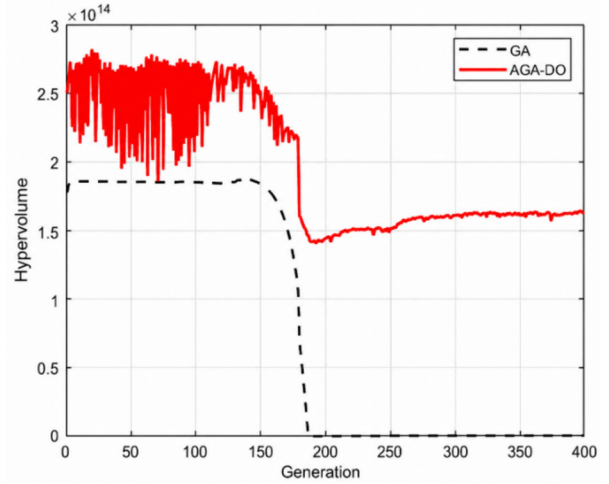


Fig.2: Learning Validity (Phase A).

4.5 Phase A: Baseline Algorithm Validation

Phase A aims to establish a baseline understanding of algorithm behavior under static and mildly dynamic conditions. The objective of this phase is not to demonstrate superiority under extreme conditions.

Fig. 2 presents the hypervolume (HV) convergence under a baseline dynamic energy scenario.

5. EXPERIMENTAL DESIGN: PROGRESSIVE VALIDATION FRAMEWORK

A five-phase progressive validation framework (Phase A–E) is employed to evaluate learning, adaptability, robustness, and recovery under increasing environmental complexity, as summarized in Table 4. All experiments were independently repeated 10 times using random seeds 1–10. MATLAB R2024a was used on an Intel Core i7 workstation with 32

GB RAM. Identical population sizes, generation limits, stopping criteria, and random-seed policies were applied to all algorithms, with no algorithm-specific parameter retuning across phases.

The results confirm that AGA-DO achieves faster convergence and significantly lower variance compared to the standard GA.

As shown in **Fig. 3**, the number of active operators decreases progressively during optimization. AGA-DO progressively removes ineffective operators while preserving a minimal effective set, achieving a balance between exploration and exploitation in dynamic environments.

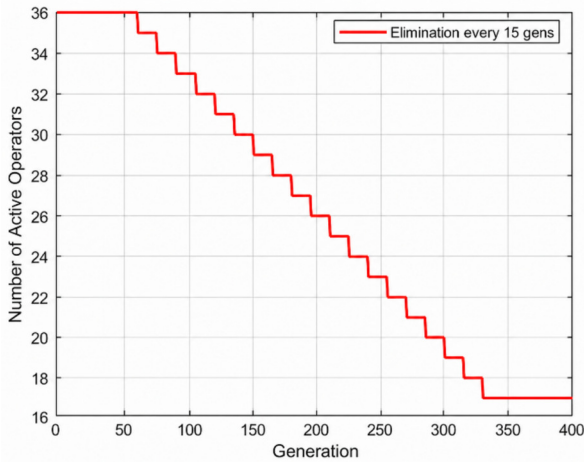


Fig.3: Progressive Reduction of The Active Composite Operator Pool From 36 to Minimum Active Size Of 17.

Table 5 summarizes the common experimental settings used across all algorithms, including population size, generation limit, operator configuration, crossover rate, and termination criterion.

5.1 Phase B: Comparison with Adaptive Genetic Algorithms

Phase B distinguishes learning-driven adaptation from conventional parameter-level adaptation. In this phase, AGA-DO is compared with an adaptive genetic algorithm that adjusts control parameters

Table 5: Experimental settings for all algorithms.

No	Parameters	Value
1	Population Size	40
2	Number of Generations	400, 500 (Phase E)
3	Crossover operators	6
4	Mutation operators	6
5	Crossover rate	0.8
6	Termination Criterion	Maximum Generations Reached

such as mutation and crossover rates during evolution.

The baseline algorithm parameters used in this study are summarized in Table 6.

Table 6: Baseline algorithm parameter settings.

Algorithm	Parameters	Value
PSO	Inertia Weight (w)	0.7
PSO	Cognitive Coefficient (c1)	1.5
PSO	Social Coefficient (c2)	1.5
DE	Mutation Factor (F)	0.5
DE	Crossover Rate (CR)	0.9
NSGA-II	Crossover Probability	0.8
NSGA-II	Mutation Probability	1/n
NSGA-II	Distribution Index (c)	20
NSGA-II	Distribution Index (m)	20

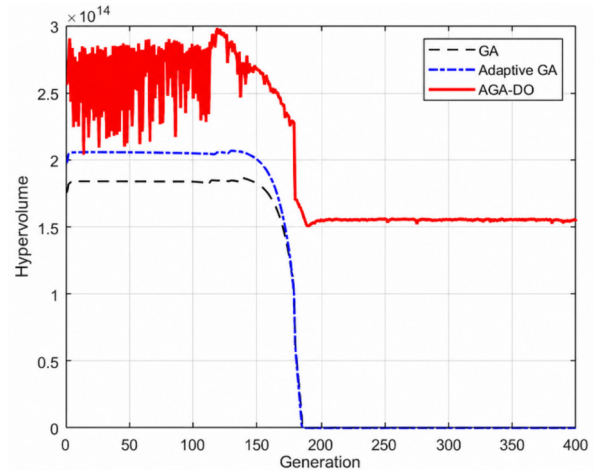


Fig.4: Comparison of Hypervolume Convergence.

Table 4: Progressive Validation Framework (Phase A–E).

Phase	Focus	Scenario Type	Key Objective	Main Evidence
Phase A	Baseline Validation	Dynamic Energy Scheduling	Verify Effectiveness vs GA	Pareto Front, HV
Phase B	Adaptive Comparison	Same Scenarios as Phase A	Compare with Adaptive GA	HV Convergence
Phase C	Cross-algorithm Benchmark	Hard Scenarios	Robustness vs PSO, DE, NSGA-II	HV, Stability
Phase D	Ablation Study	Single Hardest Scenario	Validate Necessity of AGA-DO Components	Recovery Gain, Memory Shift
Phase E	Real-world Validation	Real Demand & Wind Profiles	Robustness under Repeated Shocks	Recovery Loss, Stability

Fig. 4 demonstrates that AGA-DO maintains more stable convergence behavior than Adaptive GA. While Adaptive GA improves convergence speed, AGA-DO consistently achieves higher hypervolume and maintains stable learning trajectories, indicating superior adaptive intelligence.

The results in **Fig. 5** show that AGA-DO provides a broader and more favorable cost–emission trade-off than GA and Adaptive GA under the tested dynamic scenario.

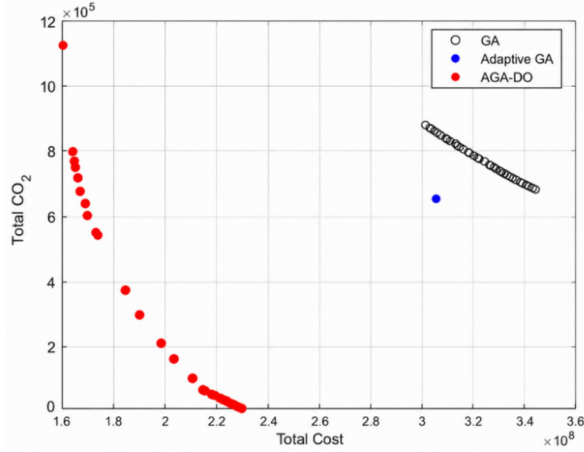


Fig.5: Pareto Front Comparison Between GA, Adaptive GA, And AGA-DO.

5.2 Phase C: Cross-Algorithm Benchmark

The environment in Phase C introduces pronounced dynamic behavior, including non-stationary objectives and constraints. This phase evaluates the general adaptability of different algorithmic paradigms and highlights the strengths and limitations of population-based versus learning-driven approaches.

As shown in **Fig. 6**, AGA-DO produces a broader and more continuous Pareto front in the raw cost–emission objective space, whereas the benchmark algorithms generate either isolated solutions or a narrower trade-off region. This indicates that AGA-DO is not limited to a single operating point, but can maintain diverse trade-off alternatives between total cost and CO₂ emission under the dynamic energy scheduling scenario. In particular, the extended front obtained by AGA-DO reflects stronger search diversity and better adaptation to the changing optimization landscape.

In **Fig. 7**, the relative hypervolume degradation is caused by a sudden system shock. AGA-DO experiences a significantly smaller performance drop than competing algorithms, reflecting intrinsic robustness or long-term performance retention to abrupt environmental changes.

Fig. 8 summarizes the variability of post-shock recovery across algorithms. AGA-DO exhibits con-

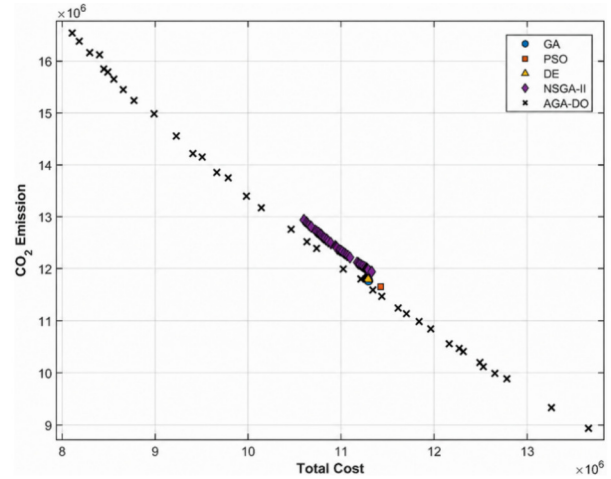


Fig.6: Final Pareto Fronts Obtained By GA, PSO, DE, NSGA-II, And AGA-DO in the Raw Cost–Emission Objective Space Under the Dynamic Energy Scheduling Scenario.

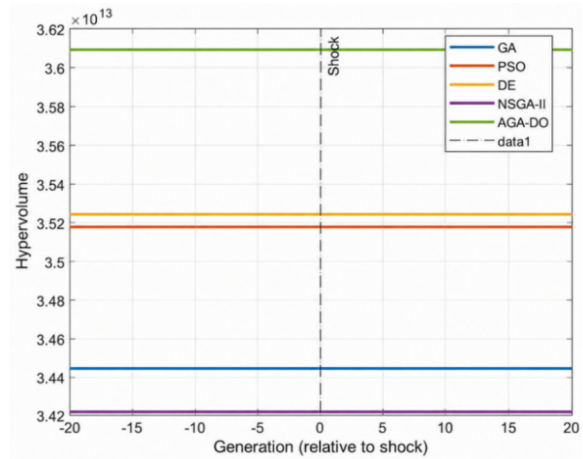


Fig.7: Shock Impact Severity (Phase C).

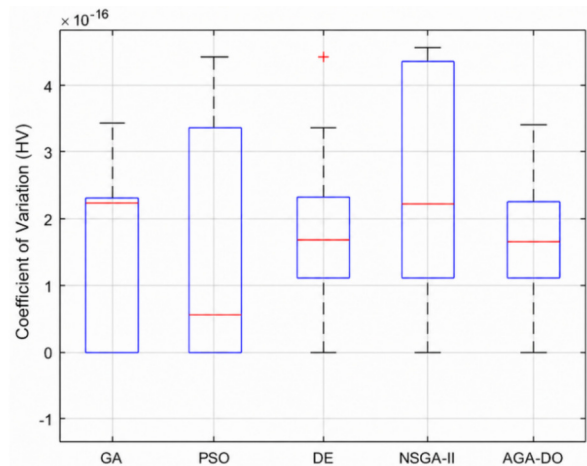


Fig.8: Recovery Stability (Phase C).

trolled recovery variability, with a distribution comparable to DE and narrower than those of GA and NSGA-II, indicating consistent post-shock behavior.

5.3 Phase D: Ablation Study

Phase D is designed to analyze the contribution of individual components within the proposed framework. By selectively disabling key mechanisms such as temporal smoothing or operator elimination, the ablation study examines their impact on learning stability and overall performance.

The results in **Fig. 9** indicate that the impact of component ablation on the cost–emission trade-off can be clearly observed.

Removing temporal smoothing or operator elimination increases short-term hypervolume but reduces learning refinement and long-term robustness.

CV denotes the coefficient of variation of post-shock recovery gain. Memory Shift represents the cumulative change in operator learning memory (EMA score) throughout the recovery period. CRL is not treated as an optimization objective but as a post hoc evaluation metric capturing accumulated adaptation inefficiency.

As shown in Table 7, the full AGA-DO configuration provides the most balanced behavior across hypervolume quality, shock response, recovery consistency, and operator-memory stability. Although NO_ELIM achieves a slightly higher mean HV, it also shows a larger HV drop and higher memory shift, indicating weaker long-term robustness. Similarly, NO_STAB exhibits a high recovery ratio but also produces a larger HV drop and greater recovery variability. These results confirm that EMA-based smoothing, operator elimination, and stability control jointly support robust operator-level learning under dynamic conditions.

5.4 Phase E: Real-World Shock and Recovery Validation

Phase E evaluates AGA-DO under a real-world, highly non-stationary environment with externally injected shocks. The “Hourly Energy Demand Generation and Weather” dataset for Spain was used; its source, coverage, variables, preprocessing, and access date are summarized in Table 8.

To make the real-world validation reproducible, the shock design used in Phase E is summarized in

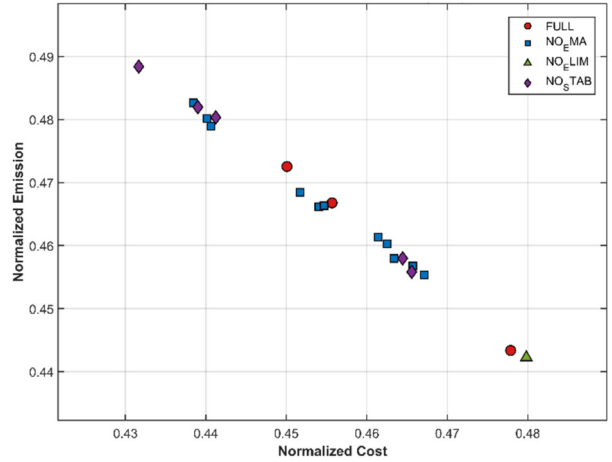


Fig.9: Normalized Cost–emission trade-off obtained from the ablation study, showing the impact of removing EMA smoothing, operator elimination, and stability control on solution quality and recovery behavior.

Table 8: Real-World Dataset Used in Phase E.

Item	Description
Dataset Name	Hourly Energy Demand Generation and Weather
Platform	Kaggle Public Dataset
Country	Spain
Resolution	Hourly
Data Coverage	1 January 2015 00:00 CET – 31 December 2018 23:00 CET
Electricity Data Source	ENTSO-E
Weather Data Source	Open Weather API
Price Data Source	Red Eléctrica De España
Access Date	29 August 2026
Fields Used	Total Load, Wind Generation, Solar Generation, Generation by Type, Price, Weather Variables
Use in this Study	Construction of Real-World Dynamic Energy Profiles
Preprocessing	Missing-Value Handling, Hourly Alignment, Normalization, Scenario Extraction
Dataset Identifier	nicholasjhana/energy-consumption-generation-prices-and-weather

Table 7: blation Study Results of AGA-DO Components.

Variant	Removed Component	Mean HV ($\times 10^{12}$)	HV Drop (%)	Recovery Ratio(%)	CV	Memory Shift
FULL	None	1.49	-0.23	108.6	0.0057	1.423
NO_EMA	Temporal Smoothing	1.28	-0.11	107.6	0.0047	1.177
NO_ELIM	Operator Elimination	1.51	-0.32	112.0	0.0057	1.920
NO_STAB	Stability Control	1.45	-0.52	114.6	0.0081	1.435

Table 9: Common Settings of The Real-World Multi-Shock Validation Scenario.

Item	Description
Scenario	Real-World Multi-Shock Validation Scenario in Phase E
Shock Structure	Three sequential shocks were imposed within the same optimization run
Common Shock Duration	80 generations per shock
Common Recovery Window	45 generations
Shock Intensity Function	$Z = \exp(-\tau/(L/3))$, where τ is the number of generations after the shock trigger and L is the shock duration
Parameter Retuning	No algorithm-specific parameter retuning was performed across phases

Table 10: Detailed Shock Events Used in The Real-World Multi-Shock Validation Scenario.

Item	Description
Scenario	Real-World Multi-Shock Validation Scenario in Phase E
Purpose	To evaluate the robustness and recovery capability of AGA-DO under repeated dynamic disturbances
Shock 1 Details	
Shock Name	Renewable Drop Shock
Trigger Generation	$0.35T_{max}$
Affected Hours	49–96
Affected Variables	PV output, wind output, net demand, ramping pressure
Shock Design	PV shortfall = 45%; wind shortfall = 38%; demand uplift = 8%; ramp-weight increase = (2.00z)
Interpretation	Represents sudden renewable generation loss, increasing dependence on dispatchable biomass generation
Shock 2 Details	
Shock Name	Severe Fuel-Emission Shock
Trigger Generation	$0.62T_{max}$
Affected Hours	73–132
Affected Variables	Fuel cost, emission factor, emission cap, biomass capacity, ramping pressure
Shock Design	Fuel-cost shock = 60%; emission pressure = 40%; emission-cap tightening = 25%; biomass capacity derating = [0.82, 0.94, 0.86]; ramp-weight increase = (1.80z)
Interpretation	Represents simultaneous economic and environmental pressure, making the cost-emission trade-off more difficult

Item	Description
Shock 3 Details	
Shock Name	Volatile Recovery Shock
Trigger Generation	$0.82T_{max}$
Affected Hours	109–156
Affected Variables	Residual demand, renewable instability, ramping pressure
Shock Design	Residual demand stress = 12%; renewable instability = 25%; oscillatory recovery pattern; ramp-stress increase = (2.50z)
Interpretation	Represents unstable post-disturbance recovery conditions after repeated shocks

Table 9. The multi-shock scenario was constructed to represent three major disturbance patterns in dynamic energy systems: renewable generation loss, severe fuel-emission pressure, and unstable recovery after repeated disruptions. All algorithms were evaluated under the same shock schedule.

Based on these common settings, three distinct disturbance events were imposed to represent renewable-generation loss, fuel-emission pressure, and unstable post-shock recovery, as detailed in Table 10. These shocks provide a controlled basis for evaluating operator-level learning and recovery under repeated real-world disturbances.

Recovery Ratio values above 100% indicate that post-shock performance exceeded the pre-shock reference level. This distinction ensures that learning quality is assessed independently of algorithmic bias toward specific performance indicators.

As shown in **Fig. 10**, AGA-DO maintains stable hypervolume performance across the three sequential shock events and demonstrates consistent post-shock recovery. Compared with the benchmark algorithms, its performance degradation is more controlled, while recovery remains sustained after repeated disturbances. These results indicate that operator-level learning enables AGA-DO to retain useful search behavior and adapt effectively under highly non-stationary real-world conditions.

Importantly, all algorithms were evaluated under the same real-world shock profiles, and no algorithm-specific parameter retuning was performed across phases.

Phase E directly assesses whether adaptive intelligence mechanisms enable sustained performance under real-world conditions (Power Demand and Renewable-Energy Dataset), providing a definitive evaluation of learning capability and resilience.

5.5 Performance Metrics and Evaluation Criteria

Algorithm performance is evaluated using hypervolume (HV), performance variance, recovery time, stability, and Cumulative Regret Loss (CRL). CRL

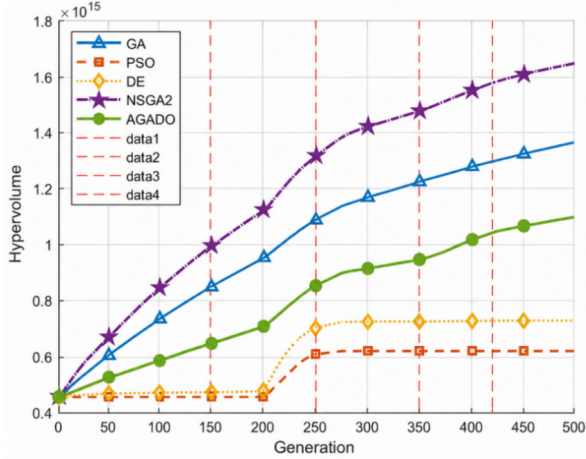


Fig.10: Hypervolume Evolution Under Multi-Shock Robustness Validation in Phase E.

is used as a post hoc observational indicator rather than an optimization objective to quantify accumulated adaptation inefficiency under repeated environmental changes. In Phase E, CRL is defined as

$$CRL = \sum_{t=1}^T (f_{best}(t) - f_{alg}(t))$$

where $f_{alg}(t)$ is the performance of the evaluated algorithm, $f_{best}(t)$ is the best reference performance at time t , and T is the evaluation horizon. Lower CRL indicates lower cumulative performance loss and more consistent adaptation. HV is used as the performance reference for CRL computation.

6. RESULTS AND DISCUSSION

This section presents and analyzes the experimental results obtained from the progressive validation framework described in Section 5. Rather than focusing solely on final objective values, the analysis emphasizes learning behavior, robustness, stability, and recovery capability, which are critical characteristics of adaptive computational intelligence in non-stationary environments.

Table 11 shows that AGA-DO achieves the lowest cumulative regret and the best overall Friedman ranking among the evaluated algorithms, indicating superior long-term adaptation and learning retention under repeated shocks. Friedman test confirmed significant performance differences among algorithms ($p < 0.05$).

Lower CRL values indicate stronger long-term robustness, faster adaptation, and reduced cumulative performance loss under repeated environmental shocks.

AGA-DO achieved the lowest mean CRL (0.81) and the best overall Friedman rank (1.2), demonstrating superior long-term adaptation under repeated environmental shocks.

Table 11: Phase E Long-Term Robustness Evaluation.

Algorithm	Mean CRL	Std. CRL	Friedman Rank
GA	91.5	93.6	4.7
DE	7.34	8.11	3.8
PSO	5.06	6.97	3.4
NSGA-II	2.13	2.42	2.1
AGA-DO	0.81	0.93	1.2

6.1 Convergence Behavior under Dynamic Conditions

In contrast, AGA-DO exhibits a more controlled convergence pattern. While its initial convergence rate may be comparable to baseline methods, it maintains the ability to adjust search direction in response to environmental changes. This behavior indicates that convergence speed alone is not a sufficient indicator of adaptive intelligence in dynamic optimization.

Table 11 highlights that AGA-DO's advantage lies in stability and learning retention rather than aggressive convergence.

6.2 Stability and Performance Variance

Performance variance across independent runs provides insight into the stability of learning mechanisms. Algorithms without explicit learning stabilization mechanisms exhibit large variance under dynamic conditions, reflecting sensitivity to noise, initialization, and transient performance fluctuations.

AGA-DO consistently demonstrates lower variance across runs, particularly in later experimental phases. This stability can be attributed to the temporal smoothing mechanism, which prevents abrupt shifts in operator selection due to short-term performance spikes.

As a result, learning trajectories remain coherent over time, supporting reliable long-term performance.

6.3 Learning Adaptability and Operator Dynamics

An important observation from the experiments is the role of operator-level learning in sustaining adaptability. In AGA-DO, operator selection probabilities evolve gradually, reflecting accumulated performance rather than instantaneous rewards. Ineffective operators are progressively suppressed, while consistently effective operators gain influence.

This behavior contrasts with adaptive parameter-based methods, where parameter adjustments do not directly address the suitability of underlying search operators. The results suggest that learning which operators to apply is more critical than how strongly to apply them, particularly in non-stationary environments.

6.4 Shock and Recovery Performance

AGA-DO demonstrates markedly faster recovery after shock events. By maintaining a refined yet flexible operator pool, the algorithm is able to reallocate search effort efficiently when the optimization landscape changes. This recovery behavior indicates that operator-level learning enables not only adaptability but also resilience to severe disruptions.

6.5 Interpretation in Terms of Adaptive Intelligence

From an adaptive-intelligence perspective, the results indicate that AGA-DO retains useful operator-level experience while remaining responsive to repeated environmental changes. This behavior is most clearly reflected by cumulative regret in Phase E: benchmark algorithms repeatedly accumulate regret after successive shocks, whereas AGA-DO maintains substantially lower CRL. Since CRL is not optimized directly but observed as a post hoc learning indicator, the results support sustained adaptation rather than short-term reactive recovery.

7. CONCLUSION

This paper presented AGA-DO, an adaptive evolutionary intelligence framework designed for dynamic and non-stationary optimization problems. By introducing operator-level learning through dynamic scoring, temporal smoothing, and operator elimination, the proposed method enables sustained adaptability, stability, and recovery in changing environments.

A progressive experimental validation framework was employed to rigorously evaluate adaptive behavior across increasing levels of environmental complexity. The results demonstrate that AGA-DO maintains robust adaptive behavior under abrupt environmental changes and real-world conditions, even though no algorithm-specific parameter retuning was performed across phases.

The findings suggest that effective computational intelligence in non-stationary environments requires learning mechanisms that operate beyond parameter adjustment. Operator-level adaptation provides a promising direction for developing resilient optimization algorithms capable of long-term deployment in dynamic settings.

Future work will explore the extension of the proposed framework to other classes of dynamic optimization problems, including domains with higher-dimensional decision spaces and more complex interaction structures. Additionally, integrating problem-aware reward design and investigating theoretical properties of operator learning dynamics represent promising avenues for further research.

AUTHOR CONTRIBUTIONS

Conceptualization, A.C. and C.K.; methodology, A.C.; software, A.C.; validation, A.C. and C.K.; formal analysis, A.C.; investigation, A.C.; data curation, A.C.; writing—original draft preparation, A.C.; writing—review and editing, A.C. and C.K.; visualization, A.C.; supervision, C.K. All authors have read and agreed to the published version of the manuscript.

References

- [1] S. Das and P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art," in *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4-31, Feb. 2011.
- [2] P. Sun, Y. Dai and X. Qiu, "Optimal Scheduling and Management on Correlating Reliability, Performance, and Energy Consumption for Multiagent Cloud Systems," in *IEEE Transactions on Reliability*, vol. 66, no. 2, pp. 547-558, June 2017.
- [3] H. Farzin, M. Fotuhi-Firuzabad and M. Moeini-Aghaie, "A Stochastic Multi-Objective Framework for Optimal Scheduling of Energy Storage Systems in Microgrids," in *IEEE Transactions on Smart Grid*, vol. 8, no. 1, pp. 117-127, Jan. 2017.
- [4] A. Choeichum, N. Keeratipranon, and C. Khemapatapan, "Multi-Fuel Allocation for Power Generation Using GA," *Journal of Reviews on Global Economics*, vol. 6, pp. 258-268, Jun. 2017.
- [5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," in *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, April 2002.
- [6] J. Wang, F. Zhang, F. Liu, and J. Ma, "Hybrid Forecasting Model-Based Data Mining and Genetic Algorithm-Adaptive Particle Swarm Optimisation: A Case Study of Wind Speed Time Series," *IET Renewable Power Generation*, vol. 10, no. 3, pp. 287-298, 2016.
- [7] P. Ghamisi and J. A. Benediktsson, "Feature Selection Based on Hybridization of Genetic Algorithm and Particle Swarm Optimization," in *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 2, pp. 309-313, Feb. 2015.
- [8] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," in *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656-667, April 1994.
- [9] S. M. Islam, S. Das, S. Ghosh, S. Roy and P. N. Suganthan, "An Adaptive Differential Evolution Algorithm With Novel Mutation and Crossover Strategies for Global Numerical Optimization,"

- in *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 482-500, April 2012.
- [10] A. Arabali, M. Ghofrani, M. Etezadi-Amoli, M. S. Fadali and M. Moeini-Aghaie, "A Multi-Objective Transmission Expansion Planning Framework in Deregulated Power Systems With Wind Generation," in *IEEE Transactions on Power Systems*, vol. 29, no. 6, pp. 3003-3011, Nov. 2014.
 - [11] B. Fakhimghanbarzadeh, H. Marzi and H. Abolghasem, "Optimization of biomass waste gasification Combined Heat and Power System," *2010 IEEE Electrical Power & Energy Conference*, Halifax, NS, Canada, pp. 1-6, 2010.
 - [12] L. Wang, C. L. Weller, D. D. Jones, and M. A. Hanna, "Contemporary Issues in Thermal Gasification of Biomass and Its Application to Electricity and Fuel Production," *Biomass Bioenergy*, vol. 32, no. 7, pp. 573-581, Jul. 2008.
 - [13] J. Ziegenhirt, T. Bartz-Beielstein, O. Flasch, W. Konen and M. Zaefferer, "Optimization of biogas production with computational intelligence a comparative study," *IEEE Congress on Evolutionary Computation*, Barcelona, Spain, pp. 1-8, 2010.
 - [14] T. Senjyu, D. Hayashi, N. Urasaki and T. Funabashi, "Optimum configuration for renewable generating systems in residence using genetic algorithm," in *IEEE Transactions on Energy Conversion*, vol. 21, no. 2, pp. 459-466, June 2006.
 - [15] W. Xiang, B. S. Sharif, and O. R. Hinton, "An Improved Resource Allocation Scheme for Plane Code Multiple Access Using Genetic Algorithm," in *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 1, pp. 74-81, Feb. 2005.
 - [16] M.-D. Yang, Y.-P. Chen, Y.-H. Lin, Y.-F. Ho, and J.-Y. Lin, "Multiobjective optimization using nondominated sorting genetic algorithm-II for allocation of energy conservation and renewable energy facilities in a campus," *Energy Build.*, vol. 122, pp. 120-130, Jun. 2016.
 - [17] Y. -Y. Hong and R. -C. Lian, "Optimal Sizing of Hybrid Wind/PV/Diesel Generation in a Stand-Alone Power System Using Markov-Based Genetic Algorithm," in *IEEE Transactions on Power Delivery*, vol. 27, no. 2, pp. 640-647, April 2012.
 - [18] J. Zhang, H. S. -H. Chung and W. -L. Lo, "Clustering-Based Adaptive Crossover and Mutation Probabilities for Genetic Algorithms," in *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 326-335, June 2007.
 - [19] M. Zhao, Z. Chen, and F. Blaabjerg, "Optimization of Electrical System for Offshore Wind Farms via Genetic Algorithm," *IET Renewable Power Generation*, vol. 3, no. 2, pp. 205-216, Jun. 2009.
 - [20] T. Celik, "Change Detection in Satellite Images Using a Genetic Algorithm Approach," in *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 2, pp. 386-390, April 2010.
 - [21] A. S. Wu, H. Yu, S. Jin, K. . -C. Lin and G. Schiavone, "An incremental genetic algorithm approach to multiprocessor scheduling," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 9, pp. 824-834, Sept. 2004.
 - [22] S. Ganguly and D. Samajpati, "Distributed Generation Allocation on Radial Distribution Networks Under Uncertainties of Load and Generation Using Genetic Algorithm," in *IEEE Transactions on Sustainable Energy*, vol. 6, no. 3, pp. 688-697, July 2015.
 - [23] J. Mendoza, R. Lopez, D. Morales, E. Lopez, P. Dessante and R. Moraga, "Minimal loss reconfiguration using genetic algorithms with restricted population and addressed operators: real application," in *IEEE Transactions on Power Systems*, vol. 21, no. 2, pp. 948-954, May 2006.
 - [24] H. Lu and G. G. Yen, "Rank-Density-Based Multiobjective Genetic Algorithm and Benchmark Test Function Study," in *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 325-343, Aug. 2003.
 - [25] R. Zhang and J. Tao, "A Nonlinear Fuzzy Neural Network Modeling Approach Using an Improved Genetic Algorithm," in *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5882-5892, July 2018.
 - [26] S. Y. Yuen and C. K. Chow, "A Genetic Algorithm That Adaptively Mutates and Never Revisits," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 454-472, April 2009.
 - [27] Z. Wang, Q. Zhang, A. Zhou, M. Gong and L. Jiao, "Adaptive Replacement Strategies for MOEA/D," in *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474-486, Feb. 2016.
 - [28] J. A. Vrugt, B. A. Robinson and J. M. Hyman, "Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 243-259, April 2009.
 - [29] R. L. Haupt, "Phase-only adaptive nulling with a genetic algorithm," in *IEEE Transactions on Antennas and Propagation*, vol. 45, no. 6, pp. 1009-1015, June 1997.
 - [30] A. G. Bakirtzis, P. N. Biskas, C. E. Zoumas and V. Petridis, "Optimal power flow by enhanced genetic algorithm," in *IEEE Transactions on Power Systems*, vol. 17, no. 2, pp. 229-236, May 2002.
 - [31] Y. Hashimoto, H. Ishibuchi and Y. Kasai, "Genetic Algorithms for a Scheduling Problem of

- Parcel Delivery and Pickup by Drones,” in *IEEE Access*, vol. 13, pp. 12345–12356, 2025.
- [32] Y. Zhang, P. Liu and X. Wang, “Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network,” in *IEEE Access*, vol. 7, pp. 88900–88910, 2019.
- [33] I. F. Da Silva, T. B. L. Ribeiro, L. S. Santos and A. R. Pinto, “Genetic Algorithm-Based Optimization Framework for Offshore Wind Farm Layout Design,” in *IEEE Access*, vol. 13, pp. 45678–45689, 2025.
- [34] Z. Qin, D. Osowski, K. Gao and L. Song, “The Research on Demand-Based Regulation of Mine Airflow Based on Niche Genetic Algorithm,” in *IEEE Access*, vol. 13, pp. 34567–34578, 2025.
- [35] S. H. Shoffan and R. Djenouri, “GA-UNet: Genetic Algorithm-Optimized Lightweight U-Net Architecture for Multi-Sequence Brain Tumor MRI Segmentation,” in *IEEE Access*, vol. 13, pp. 23456–23467, 2025.
- [36] H. Mahmoud, K. Seddik and S. Khaddaj, “TempGA: A Temperature-Inspired Adaptive Genetic Algorithm for Solving 7DOF Inverse Kinematics Problems,” in *IEEE Access*, vol. 13, pp. 56789–56800, 2025.



Anurak Choeichum is a Ph.D. candidate in Computer Engineering at Dhurakij Pundit University (DPU), Bangkok, Thailand. He received his bachelor's and master's degrees in Computer Engineering from DPU. He is currently with the Provincial Electricity Authority (PEA), Thailand, where he serves as Head of the Planning Section in the Operations Division, PEA Area 1 (Central Region). His professional experience

includes SCADA systems, power-quality systems, smart substations, IoT, automation, and digital energy applications. He has authored and co-authored more than 40 research publications in national and international journals and conferences. His research interests include evolutionary computation, multi-objective optimization, machine learning, smart grids, energy-system optimization, automation, and intelligent engineering applications. He has received several engineering, innovation, research, and employee awards from PEA and international conferences.



Chaiyaporn Khemapatapan received his B.Eng. in Electrical Engineering from King Mongkut's Institute of Technology North Bangkok in 1994, M.Eng. in Electrical Engineering from Chula-longkorn University in 1997, and Ph.D. in Electrical Engineering from Chulalongkorn University in 2005. In 2004, he was a Research Student at the Mobile Communication Research Group (ARAKI LAB), Tokyo Institute

of Technology, Japan, supported by the Thailand Research Fund (TRF) under the Royal Golden Jubilee (RGJ) Program. Dr. Khemapatapan has an extensive career in academia spanning over two decades. Currently, he serves as the Dean of the College of Engineering and Technology and the Director of the Computer Engineering Graduate Program at Dhurakij Pundit University. His research expertise encompasses Computer Networking, Mobile and Wireless Networks, Cyber Security, Cloud Computing, Machine Learning, AI and Quantum Computing. Dr. Khemapatapan has authored and co-authored numerous inter-national academic papers, with recent works focusing on transaction signing processes, multi-cloud storage security, DDoS attack defense in SDN, genetic algorithms, and smart grid distribution. Dr. Khemapatapan's blend of academic excellence and engineering management experience positions him as a key figure in engineering education and research in Thailand, particularly in the fields of computer engineering.