Constrained Optimal Pulse-Width Modulation Using Dual-Population Differential Evolution

Shengqi Gui*, Zusheng Tan†, Jing-Yu Ji*, and Billy Chiu†
*College of Management, Shenzhen University, Shenzhen, China
shengqigui0301@gmail.com
†School of Decision Science, Lingnan University, Hong Kong

***Abstract*—Synchronous optimal pulse-width modulation (PWM) is an effective means of improving multilevel inverter operation, thereby supporting steady and uninterrupted power delivery in consumer-electronics applications. The associated design task is inherently difficult because PWM optimization is commonly formulated with coupled nonlinear equality and inequality constraints, under which many classical optimizers become unreliable. In this paper, we develop a cooperative constrained evolutionary algorithm with two interacting populations. The first population is primarily guided by a constraint-handling technique to rapidly approach feasibility, while the second population coevolves to preserve search diversity and assist exploration. After the feasible region is sufficiently identified, the first population shifts its emphasis toward objective minimization and adopts an alternative constraint-handling strategy. The two populations are linked through an elite-offspring selection mechanism so that information exchange promotes coordinated progress. The proposed method is assessed in two stages. We first validate both overall performance and the benefit of the cooperative mechanism on 18 standard constrained benchmarks. We then apply the algorithm to practical synchronous optimal PWM design for multilevel inverters with different numbers of voltage levels. Statistical comparisons show that the proposed approach consistently outperforms several state-of-the-art algorithms in producing high-quality PWM control solutions.**

***Index Terms*—Synchronous optimal pulse-width modulation, constrained optimization, dual-population differential evolution.**

I. Introduction

Synchronous optimal PWM is commonly developed to meet two key requirements. First, the switching pattern must satisfy topological conditions needed for soft-switching operation [1]. Second, the method should suppress undesirable harmonics [2], since harmonic components can increase thermal stress and braking torque, leading to higher energy loss, reduced efficiency, and shortened lifetime.

To address these requirements, a variety of deterministic methods, including the Newton–Raphson scheme and Lagrangian relaxation, have been applied to reduce total harmonic distortion or to selectively eliminate specified harmonic orders. In practice, however, the success of such approaches often depends strongly on carefully chosen initial guesses, which are difficult to obtain beforehand. Moreover, many formulations require smooth objectives (e.g., twice differentiability), further restricting their use. Because PWM optimization is influenced by diverse physical factors, such as circuit topology, the number of switching devices, and phase-angle configurations, the resulting search landscapes are frequently multimodal, nonsmooth, and nonconvex. Consequently, conventional solvers may converge prematurely to local optima.

For heuristic optimization, dedicated constraint-handling techniques (CHTs) [3]–[5] are typically introduced to enforce the feasibility-related requirement. Broadly, widely used CHTs can be grouped into four families: 1) penalty-function approaches [6]; 2) feasibility rule-based methods [7]; 3) ε-constrained strategies [8]; and 4) multiobjective reformulations [9]. Penalty methods merge constraints and objective into a single unconstrained target via penalty coefficients, which makes implementation straightforward but introduces parameter sensitivity. Feasibility rules avoid external parameters by ranking solutions according to overall constraint violation and objective value; in many variants, greedy selection is used to accelerate movement toward feasibility. The ε-constrained paradigm relies on adaptively adjusting an ε threshold so that mildly infeasible candidates can be treated as feasible during evolution. Multiobjective formulations, in contrast, cast constraints and objective(s) as multiple objectives, where globally feasible and competitive solutions can emerge as nondominated points on the corresponding Pareto front.

Beyond feasibility, minimizing current harmonic distortion remains crucial for stable consumer-electronics power delivery. Nonetheless, premature convergence, a well-known issue in deterministic optimizers, can also arise in heuristic searches, leading to stagnation around infeasible candidates or locally feasible solutions [10]–[12]. To mitigate this, diversity-preservation mechanisms [13]–[15] are often incorporated to strengthen exploration, even before the search enters the feasible region.

Although existing CHTs differ in their selection logic and parameterization, most primarily concentrate on nonlinear equality/inequality constraint satisfaction. Additional consid-

This research was supported by the Scientific Research Foundation for the Phase III Construction of a High-Level University for Youth Scholars at Shenzhen University (Grant No. 000001032933), by the Interdisciplinary Team Research Project of the College of Management, Shenzhen University (Grant No. 20240408), and by the Shenzhen Key Talent Research Start-up Grant Programme (Grant No. 8270001192). This work was done partly when Mr. Gui was with College of Management, Shenzhen University. He is now with the School of Engineering, The University of Edinburgh, Edinburgh, United Kingdom. *(Corresponding author: Jing-Yu Ji)*

erations are therefore required to enhance objective optimization, especially under difficult landscapes. Furthermore, it is desirable to develop a generalized synchronous optimal PWM optimization framework that can be applied across inverter levels and can benefit from heuristic methods suited to black-box problem settings.

Motivated by these observations, we propose a cooperative dual-population constrained differential evolution approach, termed DC-DE. The remainder of this paper is organized as follows. Section II describes DC-DE in detail. Section III gives the adopted CHTs. Section IV presents experimental studies and analyses. Section V concludes the paper.

II. Constraint-Handling Techniques

To cope with frequent infeasible candidates in constrained optimization, evolutionary algorithms often employ dedicated CHTs. This section outlines four representative CHT families used in this work.

A. Penalty Function Method

Penalty methods [6], [16] merge constraint violations into the objective so that infeasible solutions receive an extra cost:

$$\text{Minimize} \quad \psi(\mathbf{x}) = f(\mathbf{x}) + \omega_0 \sum_{j=1}^{q} \omega_j \big(G_j(\mathbf{x})\big)^{\beta}, \tag{1}$$

where ω_0, ω_j, and β are penalty parameters. The violation of the jth constraint is

$$G_j(\mathbf{x}) = \begin{cases} \max\{g_j(\mathbf{x}), 0\}, & j = 1, \ldots, p, \\ \max\{|h_j(\mathbf{x})| - \delta, 0\}, & j = p+1, \ldots, q, \end{cases} \tag{2}$$

with $\delta > 0$ being the equality tolerance. A known limitation is sensitivity to penalty coefficients; therefore, adaptive/self-adaptive variants are often adopted to reduce manual tuning.

B. Feasibility Rule

The feasibility rule [17]–[19] is a parameter-free selection principle based on the overall normalized violation

$$C(\mathbf{x}) = \sum_{j=1}^{q} \frac{G_j(\mathbf{x})}{G_j^{\max}}, \tag{3}$$

where $G_j^{\max}$ is the largest observed violation for constraint j. Given two solutions: (i) any feasible solution is preferred over any infeasible one; (ii) between feasible solutions, the one with smaller $f(\mathbf{x})$ is preferred; (iii) between infeasible solutions, the one with smaller $C(\mathbf{x})$ is preferred. This rule promotes rapid feasibility attainment but may become overly greedy and induce premature convergence.

C. ε-Constrained Method

The ε-constrained method [8], [20], [21] relaxes strict feasibility preference by treating solutions with $C(\mathbf{x}) \leq \varepsilon$ as conditionally feasible. The threshold is reduced over the evaluation budget:

$$\begin{cases} \varepsilon = \varepsilon_0 \left(1 - \frac{UFEs}{MaxFEs}\right)^{\lambda}, \\ \lambda = \frac{-\log_{10}\delta - \log_{10}\varepsilon_0}{\log_{10} 0.1}, \end{cases} \tag{4}$$

where ε_0 is typically initialized by the maximum observed $C(\mathbf{x})$. Selection follows: compare by $f(\mathbf{x})$ if both satisfy $C(\cdot) \leq \varepsilon$; otherwise prefer the one satisfying the threshold; if neither satisfies, prefer smaller $C(\cdot)$.

D. Multiobjective Constraint Handling

A common multiobjective reformulation [9], [22], [23] treats constraint violation as an additional objective:

$$\text{Minimize} \quad B(\mathbf{x}) = \big(f(\mathbf{x}), C(\mathbf{x})\big). \tag{5}$$

Pareto-based selection can retain diverse candidates, including mildly infeasible solutions with good objective values. However, when feasibility is difficult, additional feasibility bias or adaptive thresholds may be needed to provide sufficient pressure toward constraint satisfaction.

III. Proposed Approach

DC-DE is a dual-population constrained differential evolution framework that proceeds in two stages: (i) feasibility attainment and (ii) objective refinement within the feasible set. The two populations employ different CHTs so that feasibility pressure and population diversity can be balanced throughout evolution.

A. Stage I: Constraint Satisfaction

In the first stage, DC-DE targets rapid entry into feasible regions while maintaining sufficient diversity for exploration. Two populations, *mPop* and *aPop*, are evolved in parallel:

- *mPop* **(feasibility-driven):** uses the feasibility rule to prioritize individuals with smaller overall constraint violation, thereby accelerating feasibility discovery.
- *aPop* **(diversity-driven):** adopts an ε-constrained multiobjective formulation with objectives $(f(\mathbf{x}), C(\mathbf{x}))$ to preserve diversity and mitigate premature convergence before feasibility is reached.

1) Dominance Relation in aPop: To incorporate the ε-constrained principle into nondominated sorting, we define an ε-based dominance relation for two solutions $\boldsymbol{\mu}$ and $\boldsymbol{\upsilon}$:

1) If $C(\boldsymbol{\mu}) \leq \varepsilon$ and $C(\boldsymbol{\upsilon}) \leq \varepsilon$, compare $(f(\mathbf{x}), C(\mathbf{x}))$ using standard Pareto dominance.
2) If $C(\boldsymbol{\mu}) \leq \varepsilon$ and $C(\boldsymbol{\upsilon}) > \varepsilon$, then $\boldsymbol{\mu}$ dominates $\boldsymbol{\upsilon}$ (and vice versa).
3) If $C(\boldsymbol{\mu}) > \varepsilon$ and $C(\boldsymbol{\upsilon}) > \varepsilon$, the individual with smaller $C(\cdot)$ dominates.

The threshold ε expands the region conditionally regarded as feasible, allowing multiobjective selection to act early and maintain a well-spread population as ε decreases gradually across generations.

2) Cooperative Constraint Handling: During constraint satisfaction, DC-DE applies *cross-selection* so that each population can exploit useful offspring generated by the other. In each generation, offspring populations Q_1 and Q_2 are generated from *mPop* and *aPop*, respectively. The two parent populations are then updated using their own CHTs, but based on pooled information:

Algorithm 1: Stage I: Constraint Satisfaction

Input: *PS*: population size; *mPop*, *aPop*: two parent populations; $UFEs$: used FEs.
Output: Updated *mPop* and *aPop*.

Offspring generation and parameter update:
$Q_1 \leftarrow$ OffspringGen(*mPop*); $Q_2 \leftarrow$ OffspringGen(*aPop*);
$UFEs \leftarrow UFEs + 2PS$;
Update ε using (4);

Update *mPop* **(feasibility rule with cross-selection):**
for $i = 1$ **to** *PS* **do**
  $mPop[i] \leftarrow \text{Best}_{FR}(mPop[i], Q_1[i], Q_2[i])$;
end

Update *aPop* **(ε-dominance + nondominated truncation):**
$\mathscr{U} \leftarrow aPop \cup Q_1 \cup Q_2$;
$aPop \leftarrow \text{NDS\_Trunc}(\mathscr{U}, PS, \varepsilon\text{-Dom})$;

Algorithm 2: Stage II: Objective Optimization

Input: *PS*: population size; *mPop*, *aPop*: two parent populations; $UFEs$: used FEs.
Output: Best solution $\mathbf{x}_{best}$ from *mPop* and *aPop*.

Offspring generation and parameter update:
$Q_1 \leftarrow$ OffspringGen(*mPop*); $Q_2 \leftarrow$ OffspringGen(*aPop*);
$UFEs \leftarrow UFEs + 2PS$;
Update ε using (4);

Update *mPop* **(scalar refinement):**
for $i = 1$ **to** *PS* **do**
  if $\phi(Q_1[i]) < \phi(mPop[i])$ **then**
    $mPop[i] \leftarrow Q_1[i]$;
  end
end

Update *aPop* **(diversity-preserving selection):**
$\mathscr{U} \leftarrow aPop \cup Q_2$;
$aPop \leftarrow \text{NDS\_Trunc}(\mathscr{U}, PS, \varepsilon\text{-Dom})$;
$\mathbf{x}_{best} \leftarrow \text{BestFeasible}(mPop \cup aPop)$;

- *mPop* selects *PS* individuals from $\{mPop, Q_1, Q_2\}$ using the feasibility rule.
- *aPop* selects *PS* individuals from $\{aPop, Q_1, Q_2\}$ by nondominated sorting and pruning [24] under the ε-based dominance in Section III-A1.

This cooperative update is summarized in **Algorithm 1**.

B. Stage II: Objective Optimization

Once at least one feasible solution is found, DC-DE switches to objective optimization within the feasible region. The two populations are assigned complementary roles, since the global feasible optimum may be located in the interior or close to the feasibility boundary.

1) Population Roles and Selection:

- *mPop* **(boundary-oriented refinement):** replaces the feasibility rule with a simplified scalar criterion

$$\phi(\mathbf{x}) = f(\mathbf{x}) + C(\mathbf{x}), \tag{6}$$

  where $C(\mathbf{x})$ is computed by (3). Here, $f(\mathbf{x})$ drives objective improvement, while $C(\mathbf{x})$ discourages drifting away from feasibility and helps refine solutions near feasible boundaries.
- *aPop* **(diverse feasible exploration):** continues the ε-constrained multiobjective search on $(f(\mathbf{x}), C(\mathbf{x}))$, preserving diversity and exploring feasible regions from multiple directions.

The update procedure is given in **Algorithm 2**.

Algorithm 3: DC-DE Framework

Input: *PS*: population size; $MaxFEs$: maximum allowable FEs.
Output: $\mathbf{x}_{best}$: best identified solution.

Initialization:
Generate *mPop* and *aPop* (size *PS* each) by Latin hypercube sampling;
Evaluate $f(\mathbf{x})$ and $G_j(\mathbf{x})$ $(j = 1, \ldots, q)$ for all $2PS$ solutions;
Compute $C(\mathbf{x})$ using (3); $UFEs \leftarrow 2PS$;
Initialize ε_0 in (4) as $\max C(\mathbf{x})$ over the initial $2PS$ solutions;

Offspring generation operator:
OffspringGen($\cdot$) uses DE/current-to-rand/1 (no crossover) or DE/rand-to-best/1 (binomial crossover), each selected with probability 0.5;

while $UFEs < MaxFEs$ **do**
  if *no feasible solution has been found* **then**
    Execute **Algorithm 1**;
  else
    Execute **Algorithm 2**;
  end
end
$\mathbf{x}_{best} \leftarrow \text{BestFeasible}(mPop \cup aPop)$;

2) Independent Search After Feasibility: After feasibility is reached, information exchange is disabled. Since *mPop* and *aPop* focus on different regions (boundary refinement versus interior exploration), forcing cooperation may reduce complementarity or introduce directional bias. Therefore, the two populations proceed independently to increase the likelihood of locating the global feasible optimum.

C. Overall Workflow of DC-DE

The complete procedure is summarized in **Algorithm 3**. Two populations are initialized by Latin hypercube sampling and evaluated in a black-box manner. The aggregated violation $C(\mathbf{x})$ is computed using (3), and ε_0 in (4) is set as the maximum violation among the initial $2PS$ candidates. The algorithm runs Stage I until feasibility is detected, and then permanently switches to Stage II until $MaxFEs$ is exhausted. Offspring are generated using DE operators, where DE/current-to-rand/1 (without crossover) and DE/rand-to-best/1 (with binomial crossover) are selected with equal probability.

IV. Experimental Study

A. Benchmark Suite and Evaluation Protocol

The effectiveness of DC-DE in identifying the global feasible optimum is assessed on the IEEE CEC 2010 constrained optimization benchmark set [25]. We adopt the 18 difficult test functions with decision-space dimensions of $D = 30$, which corresponds to a medium-scale search space and is comparable to the pulse-number settings encountered in synchronous optimal PWM design. These benchmarks exhibit diverse difficulties, including nonlinearity, multimodality, rotated landscapes, narrow feasible regions, and inseparable variable interactions, thereby providing a comprehensive evaluation of both the proposed algorithm and its key strategies.

For each test instance, the computational budget is fixed to $MaxFEs = 20000D$. All algorithms are executed over 25 independent runs. Performance is quantified using three indicators computed from the best feasible solutions: success rate (SR), mean objective value (MOV), and standard deviation (Std). Unless otherwise stated, SR is reported when at least one method fails to consistently locate feasible solutions.

TABLE I. EXPERIMENTAL RESULTS OBTAINED BY DC-DE AND FIVE STATE-OF-THE-ART ALGORITHMS

Ins.	DC-DE	CCMT	VMCH	DeCODE	DAS-DE	APCSH
	MOV±Std	MOV±Std	MOV±Std	MOV±Std	MOV	MOV±Std
F01	**-8.21E-01±1.46E-03**	-8.13E-01±1.05E-02 +	-7.97E-01±1.55E-02 +	-8.19E-01±3.20E-03 +	-8.20E-01 +	-8.19E-01±3.29E-03 +
F02	**-2.27E+00±3.95E-03**	-2.17E+00±3.13E-02 +	2.17E+00±7.26E-01 +	-2.23E+00±3.49E-02 +	-2.20E+00 +	-2.23E+00±3.57E-02 +
F03	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00** ≈	2.32E+05±5.99E+05 +	2.06E+01±1.31E+01 +	**0.00E+00** ≈	**0.00E+00±0.00E+00** ≈
F04	-3.30E-06±3.98E-08	2.60E-02±1.30E-01 +	9.18E-02±2.60E-01 +	**-3.33E-06±0.00E+00** −	-2.58E-06 +	**-3.33E-06±0.00E+00** −
F05	**-4.84E+02±0.00E+00**	-4.80E+02±3.76E+00 +	4.72E+02±7.90E+01 +	-4.83E+02±1.53E-01 +	**-4.84E+02** ≈	-4.84E+02±1.15E-01 +
F06	**-5.31E+02±2.55E-02**	-5.30E+02±3.28E-01 +	5.34E+02±9.02E+01 +	-5.28E+02±1.46E+00 +	-5.30E+02 +	-5.31E+02±1.58E-01 +
F07	**0.00E+00±0.00E+00**	1.59E-01±7.97E-01 +	**0.00E+00±0.00E+00** ≈	**0.00E+00±0.00E+00** ≈	1.59E-01 +	**0.00E+00±0.00E+00** ≈
F08	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00** ≈	2.56E+02±6.34E+02 +	**0.00E+00±0.00E+00** ≈	1.72E+00 +	**0.00E+00±0.00E+00** ≈
F09	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00** ≈	2.44E+13±1.53E+13 +	8.97E+00±2.32E+01 +	**0.00E-00** ≈	5.28E-01±1.46E+00 +
F10	**6.91E-01±2.68E+00**	7.66E+00±9.30E+00 +	2.11E+13±1.11E+13 +	3.13E+01±1.72E-05 +	2.36E+01 +	1.05E+00±2.45E+00 +
F11	-3.91E-04±1.67E-06	**-4.00E-04±0.00E+00** −	**-4.00E-04±0.00E+00** −	-3.92E-04±0.00E+00 −	-3.92E-04 −	-3.92E-04±3.87E-08 −
F12	-1.99E-01±1.25E-04	**-1.99E-01±0.00E+00** −	**-1.99E-01±0.00E+00** −	-1.99E-01±1.23E-06 −	**-1.99E-01** −	**-1.99E-01±0.00E+00** −
F13	-6.79E+01±3.97E-01	**-6.82E+01±3.47E-01** −	-6.69E+01±1.36E+00 +	-6.73E+01±1.60E+00 +	-6.81E+01 −	-6.79E+01±9.74E-01 +
F14	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00** ≈	2.24E+03±1.12E+04 +	**0.00E+00±0.00E+00** ≈	**0.00E+00** ≈	**0.00E+00±0.00E+00** ≈
F15	1.19E+01±9.33E+00	**0.00E+00±0.00E+00** −	1.94E+13±1.40E+13 +	2.18E+01±1.14E+00 +	1.94E+01 +	4.17E+00±1.91E+01 −
F16	1.80E-03±4.96E-03	**0.00E+00±0.00E+00** −	8.00E-01±3.46E-01 +	**0.00E+00±0.00E+00** −	**0.00E+00** −	**0.00E+00±0.00E+00** −
F17	3.23E-01±5.83E-01	1.23E+00±2.28E+00 +	6.32E+02±2.55E+02 +	4.48E-02±1.21E-01 −	3.00E+01 +	2.80E-02±9.94E-02 −
F18	**0.00E+00±0.00E+00**	**0.00E+00±0.00E+00** ≈	9.77E+03±6.91E+03 +	3.03E-06±1.29E-05 +	**0.00E+00** ≈	**0.00E+00±0.00E-00** ≈
+/−/≈	Base	8 / 5 / 5	15 / 2 / 1	10 / 5 / 3	9 / 4 / 5	7 / 6 / 5

B. Baseline Algorithms and Parameter Settings

DC-DE is compared against five representative constrained evolutionary algorithms reported in leading journals within the last five years: CCMT [26], VMCH [27], DeCODE [28], DAS-DE [29], and APCSH [16]. For all baselines, parameter configurations are set according to their original publications to ensure a fair comparison.

For DC-DE, the population size is set to $PS = 50$. Following common practice in differential evolution and consistent with [28], [30], [31], the scaling factor is randomly selected from $\{0.6, 0.8, 1.0\}$ and the crossover rate is randomly selected from $\{0.1, 0.2, 1.0\}$ at each reproduction step.

C. Comparison with State-of-the-Art Algorithms

Across the 18 test functions with $D = 30$, DC-DE exhibits the most competitive overall performance in terms of both solution quality (MOV) and robustness (Std). As summarized in Table I by the Wilcoxon rank-sum marks, DC-DE significantly outperforms VMCH on 15 out of 18 functions (15 / 2 / 1 for $+/-/\approx$). It also achieves clear advantages over DeCODE and DAS-DE, with 10 and 9 significant wins, respectively. Compared with CCMT and APCSH, DC-DE remains strongly competitive, obtaining 8 and 7 significant improvements, while the remaining instances are either statistically comparable or favor the baselines.

From a function-wise perspective, DC-DE frequently reaches the best (or statistically best) MOV with small Std on many instances, including several cases where it attains exact or near-exact optimal values (e.g., multiple functions with MOV $= 0$ and Std $= 0$). DC-DE is outperformed on a smaller subset of instances (e.g., around F11–F17), where CCMT provides lower MOV on some functions (e.g., F11 and F12), and DeCODE/APCSH achieve better results on a few cases (e.g., F04 and F16/F17). Nevertheless, DC-DE generally maintains stable performance across runs, whereas VMCH shows extremely large MOV and Std on multiple problems, indicating reduced reliability in the 30-dimensional setting. Overall, the 30D results confirm that DC-DE achieves a favorable balance between objective minimization and robustness relative to the five competing algorithms.

V. EXPERIMENTAL STUDY

A. Effectiveness of Task Switching

To examine the contribution of this task-switching mechanism, two variants without stage transition are considered: DC-DE-CS maintains collaboration throughout the entire run, whereas DC-DE-OO enforces independent evolution from initialization to $MaxFEs$.

On the 18 test functions with $D = 30$, DC-DE consistently surpasses its two no-switching variants in terms of objective quality and reliability. As summarized by the Wilcoxon signs in Table II, DC-DE is significantly better than DC-DE-CS on 16 functions and worse on only 2 (16 / 2 / 0 for $+/-/\approx$). Although DC-DE-CS attains SR$= 1.00$ on all functions, its MOV values deteriorate substantially on many instances (e.g., F03, F07–F09, and F14–F15), indicating that maintaining collaboration throughout the run tends to hinder effective objective refinement after feasibility has been reached. Compared with DC-DE-OO, DC-DE also shows clear overall advantages (14 wins, 3 losses, and 1 tie). In addition, DC-DE-OO is less reliable in feasibility attainment for difficult problems: it fails completely on F11 and F12 (SR$= 0$ with NaN MOV), and its SR drops below 1.00 on F09 and F15. While DC-DE-OO achieves slightly better MOV than DC-DE on a few instances

TABLE I. EXPERIMENTAL RESULTS OBTAINED BY DC-DE AND ITS VARIANTS

Ins.	DC-DE		DC-DE-CS		DC-DE-OO		Ins.	DC-DE		DC-DE-CS		DC-DE-OO	
	MOV	SR	MOV	SR	MOV	SR		MOV	SR	MOV	SR	MOV	SR
F01	**-8.2088E-01**	**1.00**	-8.1309E-01 +	1.00	-8.1973E-01 +	1.00	F10	6.9067E-01	1.00	1.0120E+02 +	1.00	**5.5032E-01** −	**1.00**
F02	-2.2727E+00	1.00	-2.2724E+00 +	1.00	-2.2724E+00 +	1.00	F11	**-3.9053E-04**	**1.00**	1.4343E-03 +	1.00	NaN +	0.00
F03	**0.0000E+00**	**1.00**	7.2176E+05 +	1.00	3.4408E+00 +	1.00	F12	**-1.9906E-01**	**1.00**	5.2673E+00 +	1.00	NaN +	0.00
F04	-3.2961E-06	1.00	-3.3266E-06 −	1.00	-3.3069E-06 −	1.00	F13	-6.7919E+01	1.00	**-6.8072E+01** −	**1.00**	-6.7383E+01 +	1.00
F05	**-4.8361E+02**	**1.00**	-4.8338E+02 +	1.00	-4.8360E+02 +	1.00	F14	**0.0000E+00**	**1.00**	4.8520E+01 +	1.00	1.0949E+02 +	1.00
F06	**-5.3062E+02**	**1.00**	-5.3060E+02 +	1.00	-5.3061E+02 +	1.00	F15	**1.1894E+01**	**1.00**	1.0736E+02 +	1.00	1.9469E+01 +	0.92
F07	**0.0000E+00**	**1.00**	1.2352E+01 +	1.00	4.7839E-01 +	1.00	F16	**1.8004E-03**	**1.00**	4.7351E-03 +	1.00	1.1039E-02 +	1.00
F08	**0.0000E+00**	**1.00**	4.3022E+01 +	1.00	5.4829E-01 +	1.00	F17	3.2320E-01	1.00	1.1265E+00 +	1.00	2.0838E-01 −	1.00
F09	**0.0000E+00**	**1.00**	2.1423E+02 +	1.00	1.4809E+01 +	0.96	F18	**0.0000E+00**	**1.00**	6.3214E-01 +	1.00	**0.0000E+00** ≈	**1.00**

TABLE III. DC-DE AND ECO-HCT, AGM-DE , IUCDE, MVR-IUDE AND εMHCO ON SYNCHRONOUS OPTIMAL PWM SCENARIOS

Ins.	DC-DE	ECO-HCT	AGM-DE	IUCDE	MVR-IUDE	εMHCO
	MOV±Std (SR)	MOV±Std (SR)	MOV±Std (SR)	MOV±Std (SR)	MOV±Std (SR)	MOV±Std (SR)
SC1	**4.32E-02±6.83E-03 (100)**	8.69E-02±3.42E-02 (100)	4.92E-02±9.26E-03 (100)	9.29E-02±2.04E-02 (100)	9.27E-02±3.69E-02 (100)	7.45E-02±2.83E-02 (24)
SC2	**2.95E-02±5.82E-03 (100)**	6.37E-02±3.14E-02 (100)	6.22E-02±7.77E-03 (100)	4.92E-02±1.37E-02 (100)	4.55E-02±5.36E-03 (100)	9.20E-02±4.45E-02 (36)
SC3	2.15E-02±5.07E-03 **(100)**	4.13E-02±2.00E-02 (100)	3.75E-02±9.00E-03 (100)	4.23E-02±1.84E-02 (100)	4.94E-02±2.00E-02 (100)	**1.29E-02±3.06E-03** (32)
SC4	**3.77E-02±1.61E-02 (100)**	1.10E-01±1.14E-01 (96)	5.15E-02±1.17E-02 (100)	6.81E-02±6.25E-02 (100)	4.86E-02±7.35E-03 (100)	8.98E-02±2.35E-02 (84)
SC5	**3.04E-02±4.67E-03 (100)**	8.85E-02±1.12E-01 (88)	4.20E-02±3.05E-02 (100)	5.83E-02±3.57E-02 (100)	5.83E-02±3.29E-02 (100)	3.92E-02±3.98E-03 (100)

(e.g., F10 and F17), the broader results suggest that enforcing independence from the start slows or prevents convergence to feasibility under the fixed evaluation budget.

Overall, these comparisons support the necessity of task switching in DC-DE: collaboration is beneficial for rapid feasibility discovery, whereas independent evolution is more suitable for subsequent objective optimization in high-dimensional constrained settings.

B. Real-World Synchronous Optimal PWM Scenarios

VI. EXPERIMENTAL STUDY

Synchronous optimal PWM has recently attracted increasing attention in medium-voltage drive systems because it can lower switching frequency without sacrificing waveform quality, thereby reducing switching losses and improving inverter efficiency. In this setting, switching angles within one fundamental period are optimized to suppress current harmonic distortion while satisfying a set of nonlinear equality/inequality constraints.

To examine the practical utility of DC-DE across different inverter configurations, we consider five engineering scenarios (SC1–SC5). These cases correspond to 3-, 5-, 7-, 9-, and 11-level inverters, respectively. From an optimization perspective, these problems are treated as closed-box tasks, where the solver operates without explicitly embedding detailed circuit physics; this setting highlights the advantage of heuristic search in handling complex constraint landscapes.

DC-DE is compared with five recent algorithms tailored for real-world constrained optimization, including PWM-related applications: ECO-HCT [32], AGM-DE [9], IUCDE [33], MVR-IUDE [34], and εMHCO [20]. Each method is executed for 25 independent runs using the parameter settings recommended in the corresponding references. The quantitative results are summarized in Table III.

These five scenarios are challenging due to their constraint density and nonlinearity. Specifically, SC1–SC3 involve 25 constraints (for 3-, 5-, and 7-level inverters), whereas SC4–SC5 involve 30 constraints (for 9- and 11-level inverters). As reported in Table III, ECO-HCT does not achieve SR= 100% on SC4 and SC5, and εMHCO reaches SR= 100% only on SC5. In contrast, DC-DE attains SR= 100% on all five cases. Moreover, DC-DE provides the lowest current distortion on four of the five scenarios; the only exception is SC3, where εMHCO yields the best MOV.

Overall, the real-world PWM experiments confirm that DC-DE can simultaneously deliver reliable feasibility attainment and competitive objective minimization across multiple inverter levels, demonstrating its practicality for synchronous optimal PWM design.

VII. CONCLUSION

This paper presented DC-DE, a dual-population differential evolution method for closed-box constrained optimization problems with stage-dependent priorities. The algorithm explicitly separates the search into two phases: (i) cooperative constraint satisfaction to rapidly approach feasibility, and (ii) independent objective optimization to refine solutions within (or near the boundary of) the feasible set. Different constraint-handling techniques are assigned to the two populations to match the needs of each phase.

Behavioral analysis indicates that the two populations play complementary roles. During feasibility pursuit, cross-selection enables effective information reuse across populations, while diversity-preserving selection supports broad exploration. After feasibility is reached, independent evolution encourages balanced refinement in different regions of the feasible set.

Extensive experiments on the IEEE CEC 2010 benchmark suite and scalable synchronous optimal PWM scenarios demonstrate that DC-DE achieves strong overall performance compared with several recent state-of-the-art algorithms. These results validate both the proposed dual-population design and the stage-switching strategy as key contributors to robust global feasible optimization.

References

[1] R. Hirata and Y. Okamoto, "Reduction of memory consumption in time domain adjoint variable method for topology optimization of squirrel-cage induction motor driven by pulse width modulation inverter," *IEEJ Journal of Industry Applications*, vol. 14, no. 1, pp. 6–11, 2025.

[2] M. A. Memon, M. D. Siddique, S. Mekhilef, and M. Mubin, "Asynchronous particle swarm optimization-genetic algorithm (APSO-GA) based selective harmonic elimination in a cascaded H-bridge multilevel inverter," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1477–1487, 2022.

[3] J.-Y. Ji, B. Chiu, Z. Tan, B. Niu, L. Liang, and S. Kwong, "Surrogate-assisted evolutionary algorithm for expensive optimization with equality and inequality constraints," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2025.

[4] C. A. C. Coello, "Constraint-handling techniques used with evolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '22. New York, NY, USA: Association for Computing Machinery, 2022, pp. 1310–1333.

[5] J. Zhang, J.-Y. Ji, and S. Kwong, "A surrogate-assisted differential evolution for expensive optimization with equality constraints," *Applied Soft Computing*, vol. 189, p. 114547, 2026.

[6] X. Shi and C. Sun, "Penalty function-based distributed primal-dual algorithm for nonconvex optimization problem," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 2, pp. 394–402, 2025.

[7] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[8] C. Zhang, A. K. Qin, W. Shen, L. Gao, K. C. Tan, and X. Li, "ε-constrained differential evolution using an adaptive ε-level control method," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 769–785, 2022.

[9] J.-Y. Ji, Z. Tan, M.-L. Wong, and J. Zhang, "Heat-pipe-constrained IoT device layout via multiobjective differential evolution," *IEEE Internet of Things Journal*, vol. 12, no. 7, pp. 8261–8275, 2025.

[10] H. Zou, S. Zeng, C. Li, and J. Ji, "A survey of machine learning and evolutionary computation for antenna modeling and optimization: Methods and challenges," *Engineering Applications of Artificial Intelligence*, vol. 138, p. 109381, 2024.

[11] X. Yang, R. Wang, K. Li, and H. Ishibuchi, "Meta-black-box optimization for evolutionary algorithms: Review and perspective," *Swarm and Evolutionary Computation*, vol. 93, p. 101838, 2025.

[12] C.-L. Wang, Z.-J. Wang, Z.-F. Xue, Z.-H. Zhan, S. Kwong, and J. Zhang, "Fuzzy adaptive multitask optimization," *IEEE Transactions on Cybernetics*, vol. 56, no. 1, pp. 107–121, 2026.

[13] J.-Y. Ji, W.-J. Yu, W.-N. Chen, Z.-H. Zhan, and J. Zhang, "Solving multimodal optimization problems through a multiobjective optimization approach," in *2017 Seventh International Conference on Information Science and Technology (ICIST)*, 2017, pp. 458–463.

[14] T. Xu, J. He, and C. Shang, "Helper and equivalent objectives: Efficient approach for constrained optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 1, pp. 240–251, 2022.

[15] J.-Y. Ji and M. L. Wong, "Surrogate-assisted parameter re-initialization for differential evolution," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 1592–1599.

[16] J. Li, G. Li, Z. Wang, and L. Cui, "Differential evolution with an adaptive penalty coefficient mechanism and a search history exploitation mechanism," *Expert Systems with Applications*, vol. 230, p. 120530, 2023.

[17] E. Mezura-Montes, C. A. Coello Coello, and E. I. Tun-Morales, "Simple feasibility rules and differential evolution for constrained optimization," in *MICAI 2004: Advances in Artificial Intelligence*, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 707–716.

[18] B. Sun, W.-J. Yu, X.-F. Liu, J. Zhong, J.-Y. Li, Z.-H. Zhan, S. Kwong, and J. Zhang, "Tensor-based ant colony optimization for set meal design in online-to-offline restaurants," *IEEE Transactions on Cybernetics*, pp. 1–14, 2026.

[19] J.-Y. Ji, Z. Tan, S. Zeng, and M.-L. Wong, "An ε-constrained multiobjective differential evolution with adaptive gradient-based repair method for real-world constrained optimization problems," *Applied Soft Computing*, vol. 152, p. 111202, 2024.

[20] J. Liu, J. Feng, S. Yang, H. Zhang, and S. Liu, "Dynamic ε-multilevel hierarchy constraint optimization with adaptive boundary constraint handling technology," *Applied Soft Computing*, vol. 152, p. 111172, 2024.

[21] J.-Y. Ji, S. Yang, K. Y. Lee, B. Chiu, M. L. Wong, and S. Kwong, "An improved gradient-based repair method for constrained numerical optimization," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024, pp. 1–8.

[22] R. Jiao, B. Xue, and M. Zhang, "Investigating the correlation amongst the objective and constraints in Gaussian process-assisted highly constrained expensive optimization," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 5, pp. 872–885, 2022.

[23] J. Zhang, B. Chiu, J.-Y. Ji, L. Liang, J. Zhang, and S. Kwong, "Region division and merging-based multiobjective optimization for multimodal problems," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 3, pp. 7651–7661, 2025.

[24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[25] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2010 competition on constrained real-parameter optimization," *Nanyang Technological University, Singapore*, vol. 24, p. 910, 2010.

[26] Y. Li, W. Gong, Z. Hu, and S. Li, "A competitive and cooperative evolutionary framework for ensemble of constraint handling techniques," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 4, pp. 2440–2451, 2024.

[27] G. Wu, X. Wen, L. Wang, W. Pedrycz, and P. N. Suganthan, "A voting-mechanism-based ensemble framework for constraint handling techniques," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 4, pp. 646–660, 2022.

[28] B.-C. Wang, H.-X. Li, Q. Zhang, and Y. Wang, "Decomposition-based multiobjective optimization for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 574–587, 2021.

[29] H. Huang, Y. Xu, Y. Xiang, and Z. Hao, "Correlation-based dynamic allocation scheme of fitness evaluations for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 5, pp. 1250–1264, 2024.

[30] J.-Y. Ji, Z. Tan, S. Zeng, E. W. K. See-To, and M.-L. Wong, "A surrogate-assisted evolutionary algorithm for seeking multiple solutions of expensive multimodal optimization problems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 377–388, 2024.

[31] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2016.

[32] H. Peng, Z. Xu, J. Qian, X. Dong, W. Li, and Z. Wu, "Evolutionary constrained optimization with hybrid constraint-handling technique," *Expert Systems with Applications*, vol. 211, p. 118660, 2023.

[33] L. Deng, G. Yuan, C. Li, and L. Zhang, "Differential evolution with individual and correlation information utilization for constrained optimization problems," *Information Sciences*, vol. 700, p. 121863, 2025.

[34] A. Song, G. Wu, L. Zhou, L. Wang, and W. Pedrycz, "Exact and metaheuristic algorithms for variable reduction," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 6, pp. 1704–1718, 2024.