Transformer Autoencoder-Assisted Evolutionary Framework for Constrained Multiobjective 3D Wind Farm Layout Optimization

Yuhang Ma, Shangce Gao, Anqi Pan, and Bo Shen

***Abstract*—As the global energy crisis intensifies, optimizing wind farm layouts has become increasingly important. However, most existing studies formulate the wind farm layout optimization problem (WFLOP) as a single-objective optimization problem and rely on idealized 2D wake effect models, typically assuming flat terrain while ignoring the influence of elevation and complex topography. This paper reformulates the WFLOP as a constrained multiobjective optimization problem by integrating a 3D wake model to simultaneously maximize power output and minimize noise impact under an economic budget constraint. Although evolutionary algorithms have become popular for tackling this NP-hard problem, they often struggle with large-scale WFLOP instances defined in binary search spaces. In this work, we introduce a Transformer autoencoder-assisted evolutionary framework that maps large-scale binary layouts onto a low-dimensional, real-valued, continuous, and smooth latent manifold. Specifically, during the pre-training phase, a collection of candidate layouts is fed into the Transformer autoencoder, encouraging the latent representations to capture the spatial features of the layouts. Subsequently, during the evolutionary search process, a regression branch and a metric-alignment head are added to the Transformer autoencoder model. The current layouts, together with their relative fitness values, are then fed into the Transformer autoencoder for fine-tuning, enabling the latent space to form a continuous and smooth manifold. Afterwards, learning-assisted constrained multiobjective optimization is performed on this manifold. Extensive comparisons with state-of-the-art algorithms have demonstrated the superiority of the proposed framework.**

***Index Terms*—Transformer, autoencoder, wind farm layout optimization, constrained multiobjective optimization, evolutionary algorithms.**

I. Introduction

With the accelerating depletion of conventional energy resources and the continuously increasing global electricity demand, addressing energy challenges has become increasingly imperative [1]. In line with international initiatives such as the Paris Agreement, the European Green Deal, and China's Carbon Peaking and Carbon Neutrality targets, an accelerated transition to sustainable energy sources is essential [2]. This necessity has spurred extensive research and development in renewable energy technologies, including wind power [3], solar power [4], and tidal power [5]. Among these renewable energy sources, wind energy has emerged as one of the most promising alternatives, owing to its substantial potential to reduce greenhouse gas emissions and support a reliable electricity supply [6]. By 2023, the global installed capacity of wind energy had surpassed the 1 TW milestone, demonstrating a steadily accelerating growth trend [7].

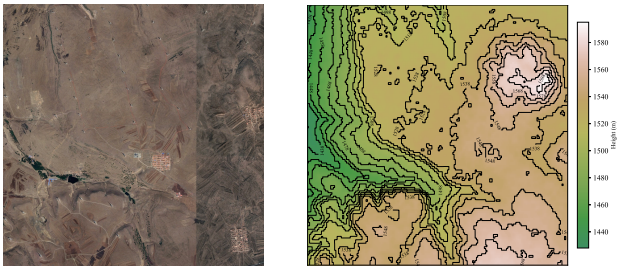
A wind farm comprises multiple wind turbines that convert wind energy into electricity. Due to the wake effect, in which downstream turbines operate within the turbulence generated by upstream turbines, power output can decrease, and mechanical stress may increase, thereby significantly degrading overall efficiency and service life [8]. Moreover, because rearranging turbines after installation is prohibitively costly, optimizing wind farm layouts before construction is essential. In recent years, discrete modeling formulations have been widely employed in studies of wind farm layout optimization problems (WFLOPs), in which the wind farm area is partitioned into a grid, and turbines are positioned at predefined grid cells [9]. Compared with continuous formulations, discrete formulations offer fewer degrees of freedom but can significantly reduce computational complexity when handling large-scale WFLOPs [10]. As a result, many studies have modeled the WFLOP as a large-scale binary optimization problem, in which a value of 1 indicates the presence of a turbine at a given location, whereas 0 indicates its absence [9].

In earlier studies, the WFLOP was typically formulated as a single-objective optimization problem aimed at maximizing energy output, often using 2D wake models. In such models, wind turbines are typically represented only by their horizontal coordinates, while the terrain is assumed to be flat. For example, Zhang et al. [11] proposed a reinforcement learning-based particle swarm optimization algorithm for turbine placement optimization, thereby improving power production. In addition, practical WFLOPs are inherently multiobjective, requiring the simultaneous optimization of factors such as energy output and noise impact. Recently, Moreno et al. [12] simultaneously considered land area, annual power generation cost, total wind farm area, and wake effect losses as optimization objectives. Zhang et al. [13] considered wind farm power output, land use, and costs. Yang et al. [14] optimized two objectives: power production and turbine fatigue life.

This work was supported in part by the National Natural Science Foundation of China under Grants 62273088 and 62303108, in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP25K03179, and in part by the Fundamental Research Funds for the Central Universities under Grant CUSF-DH-T-2023065. (*Corresponding author: Bo Shen.*)

Y. Ma is with the School of Information and Intelligent Science, Donghua University, Shanghai 201620, China, and also with the Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan. S. Gao is with the Faculty of Engineering, University of Toyama, Toyama 930-8555, Japan. A. Pan and B. Shen are with the School of Information and Intelligent Science, Donghua University, Shanghai 201620, China (e-mail: yuhang.ma@mail.dhu.edu.cn; gaosc@eng.u-toyama.ac.jp; pananqi@dhu.edu.cn; bo.shen@dhu.edu.cn).

However, as shown in Fig. 1, real-world terrain is often not flat, and elevation changes cannot be ignored. In contrast to 2D wake models, 3D wake models represent wind turbines using both horizontal coordinates and height information, thereby enabling wake interactions to be evaluated under complex terrain conditions. Compared with many previous studies, this work considers real-world factors in wind farm layout design, such as terrain features, noise impact, and budget constraints. Accordingly, in this study, the WFLOP is formulated as a constrained multiobjective optimization problem (CMOP) by integrating a 3D wake model, with the goal of simultaneously maximizing power generation capability and minimizing noise impact under an economic budget constraint.



(a) (b)

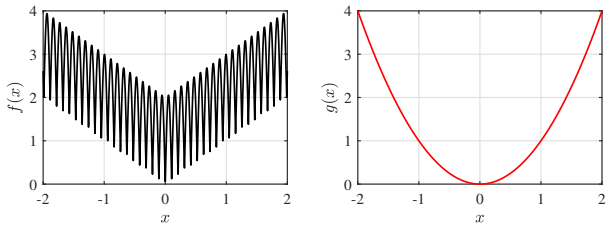
Fig. 1. Terrain information of the Zhangbei wind farm area. (a) Satellite image obtained from Google Earth. (b) Elevation contour map.

Due to the NP-hardness and nonconvexity of the WFLOP, traditional optimization algorithms often fail to yield satisfactory solutions [9]. Over the past two decades, to address these challenges, constrained multiobjective evolutionary algorithms (CMOEAs), which can produce a well-distributed set of solutions in a single run, have become increasingly popular for solving CMOPs [15], [16]. Nonetheless, evolutionary algorithms suffer from the curse of dimensionality and may become trapped in local optima when tackling large-scale optimization problems [17]. To reduce the dimensionality of large-scale WFLOPs, Lei et al. [10] proposed transforming large-scale binary optimization problems into low-dimensional integer optimization problems, where the transformed dimensionality is determined by the number of turbines and each dimension stores the index of a turbine. Although this transformation is relatively straightforward, it produces an integer-encoded search space, forming a discrete lattice rather than a continuous manifold. This discontinuity undermines the local smoothness assumed by many evolutionary operators, thereby degrading search efficiency and promoting premature convergence [18].

Autoencoder-assisted evolutionary algorithms have become increasingly popular in evolutionary computation [19], [20], [21]. By compressing the high-dimensional search space into a low-dimensional and real-valued latent space, autoencoders can significantly accelerate evolutionary search. Nonetheless, two key challenges arise when autoencoders are used to assist evolutionary search: (1) *How* to maintain high reconstruction accuracy during the evolutionary process. (2) *How* to generate a sufficiently smooth latent space that can accelerate evolutionary search.

To address the first challenge, inspired by the success of Transformer neural networks in sequence-to-sequence tasks [22], we adopt a Transformer autoencoder-based pre-training and fine-tuning paradigm. In the pre-training phase, a large set of candidate wind farm layouts is fed into the Transformer autoencoder to map the large-scale binary search space into low-dimensional and real-valued representations. Furthermore, at each evolutionary generation, we freeze the decoder and fine-tune the encoder using the current wind farm layouts to maintain reconstruction accuracy.

To clearly illustrate the second challenge, a simple example is presented in Fig. 2. Fig. 2(a) depicts a multimodal landscape, whereas Fig. 2(b) shows a smooth unimodal landscape. Although both functions have the same optimum at $(0, 0)$, the landscape in Fig. 2(b) is more conducive to evolutionary search. Hence, in addition to maintaining reconstruction accuracy, we aim to shape the latent space into a continuous and smooth manifold to accelerate evolutionary search. To this end, at each evolutionary generation, the relative fitness value of each layout in the current population is first computed using the SPEA2 fitness-assignment method [23] and the constraint dominance principle (CDP). Subsequently, a regression branch is introduced to predict the relative fitness value of each layout, and a metric-alignment head is added to encourage pairwise distances in the latent space to reflect the corresponding differences in relative fitness. As a result, small changes in latent vectors lead to correspondingly small changes in relative fitness values, thereby shaping a continuous and smooth latent landscape that facilitates subsequent learning-assisted constrained multiobjective optimization.



(a) (b)

Fig. 2. Illustration of two types of objective functions. (a) Multimodal function. (b) Smooth unimodal function.

In view of the above, a Transformer autoencoder-assisted evolutionary framework (TAAE) is proposed to address constrained multiobjective 3D WFLOPs (CMWFLOPs). In this study, TAAE integrates a basic differential evolution (DE) operator as the search engine, adopts the CDP for constraint handling, and employs NSGA-II [24] as the base multiobjective optimizer. The main contributions of this study are summarized as follows:

1) This study reformulates the WFLOP as a CMOP by incorporating a 3D wake model, aiming to simultaneously maximize power output and minimize noise impact while satisfying an economic budget constraint.
2) In the pre-training stage, a set of candidate wind farm layouts is fed into the Transformer autoencoder, guiding

its latent space to learn the spatial characteristics of the layouts.
3) In the learning-assisted optimization stage, a regression branch and a metric-alignment head are incorporated into the Transformer autoencoder model, encouraging its latent space to form a continuous and smooth manifold.
4) To ensure that each grid cell contains at most one turbine and to avoid overlaps with noise-monitoring sites, a Gaussian-based resampling strategy is proposed to repair conflicts in wind farm layouts.
5) The proposed TAAE is compared with other state-of-the-art algorithms on a set of large-scale CMWFLOPs, and the results demonstrate its superior performance in real-world scenarios.

This paper is organized as follows. Section II presents Transformer-assisted evolutionary algorithms, autoencoder-assisted evolutionary algorithms, and constrained multiobjective optimization. In Section III, the wind farm layout optimization problem is formulated as a constrained multiobjective optimization problem. Section IV describes the proposed framework. Section V describes the test instances, compared algorithms, performance metrics, and parameter settings. Section VI reports and analyzes the experimental results. Finally, Section VII concludes the paper and outlines future work.

II. Preliminaries

In this section, we briefly introduce Transformer-assisted evolutionary algorithms, autoencoder-assisted evolutionary algorithms, and constrained multiobjective optimization.

A. Transformer-Assisted Evolutionary Algorithm

The Transformer architecture, proposed by Vaswani et al. [22], was developed to overcome the limitations of conventional sequence-to-sequence models in tasks such as machine translation. Unlike traditional recurrent architectures, the Transformer dispenses with recurrence and is built primarily upon multi-head self-attention. This design enables the model to capture long-range dependencies within a sequence across multiple representation subspaces, while also supporting efficient parallel computation [25]. Currently, the Transformer serves as the backbone of many state-of-the-art models. Representative models, including BERT [26], GPT [27], and T5 [28], have demonstrated outstanding performance on numerous natural language processing tasks.

Learning-assisted evolutionary search has emerged as an active research direction in the field of evolutionary computation [29]. Recently, Dang et al. [30] attempted to utilize a Transformer model to predict promising offspring to accelerate the search process for multimodal multiobjective optimization. Nevertheless, the use of the Transformer architecture to assist evolutionary search remains in its infancy in the evolutionary computation community.

B. Autoencoder-Assisted Evolutionary Algorithm

An autoencoder is an unsupervised neural network designed to reconstruct its input, in which the number of neurons in the input layer is equal to that in the output layer [31]. In general, an autoencoder consists of two main components: (1) an encoder and (2) a decoder. The encoder compresses the input data into a lower-dimensional latent representation. This process enables the model to capture the most important features of the input while discarding redundant information. Mathematically, the encoder can be represented as a function $\vec{h} = f(\vec{x})$, where $\vec{x}$ is the input vector, and $\vec{h}$ denotes the latent representation. Conversely, the decoder is responsible for reconstructing the original input from the latent representation. This process is typically represented as $\hat{\vec{x}} = g(\vec{h})$, where $\hat{\vec{x}}$ is the reconstructed output. Typically, the encoder and decoder are jointly trained to minimize the reconstruction loss, thereby maximizing the similarity between the input $\vec{x}$ and the reconstructed output $\hat{\vec{x}}$.

In the field of evolutionary computation, autoencoders have been increasingly used to enhance the performance of evolutionary algorithms. For instance, to address large-scale multiobjective optimization problems, Liu et al. [19] developed APTEA, in which an autoencoder was used to learn the relative importance of each variable. In [20], an autoencoder combined with particle swarm optimization was designed to tackle the neural network architecture search problem. More recently, Tan et al. [21] employed an autoencoder in combination with a surrogate-assisted evolutionary algorithm to solve the vehicle lightweight design optimization problem. Despite these advances, many existing autoencoder-assisted evolutionary algorithms still struggle to maintain high reconstruction accuracy throughout the evolutionary search. Moreover, the latent space is often not explicitly regularized and may be multimodal, which can result in unsatisfactory evolutionary search performance.

C. Constrained Multiobjective Optimization

In real-world optimization scenarios, it is common to encounter complex problems that require simultaneously optimizing conflicting objectives while satisfying a variety of constraints. CMOEAs have become increasingly popular in the field of evolutionary computation [15], [16]. Existing CMOEAs can be broadly categorized into three groups: feasibility-driven algorithms, two-stage evolutionary algorithms, and coevolutionary algorithms [32]. Feasibility-driven algorithms apply the CDP by always treating feasible solutions as superior to infeasible ones. A representative example is NSGA-II-CDP [24]. Although these algorithms can rapidly identify feasible solutions, they are prone to getting trapped in local optima [33]. Two-stage evolutionary algorithms prioritize objective optimization in the initial stage and focus on constraint satisfaction in the later stage. Representative examples include PPS [34] and CMOEAS [35]. Nevertheless, determining an effective strategy for transitioning between different stages remains a significant challenge [32]. Coevolutionary algorithms employ multiple populations and accelerate the search for feasible solutions by enabling information exchange among different subpopulations. Representative algorithms include CCMO [36] and URCMO [37]. Nonetheless, collaboration among multiple populations may lead to inefficient use of computational resources [38].

III. Wind Farm Layout Optimization Problem

In this section, a constrained multiobjective 3D wind farm layout optimization problem is formulated. This problem simultaneously considers two objectives: maximizing total output power and minimizing noise impact, subject to an economic budget constraint.

A. 3D Wind Farm Energy Model

The first objective of the CMWFLOP is to maximize the total output power of the wind farm.

1) Total output power: In this study, the number of wind turbines N_w is fixed, and the total output power P_{tot} under a given wind scenario can be calculated as

$$P_{tot} = \sum_{i=1}^{N_w} \sum_{v,\theta} p(v,\theta)\, P_i(v,\theta,I), \tag{1}$$

where v and θ denote the wind speed and wind direction, respectively, $p(v,\theta)$ is their joint probability, I denotes a specific wind farm layout, and $P_i(v,\theta,I)$ is the power output of the i-th wind turbine. Since maximizing the total output power P_{tot} is desired, the first objective function f_1 can be defined as

$$f_1 = \frac{1}{P_{tot}}, \tag{2}$$

thereby converting the objective into a minimization problem.

For each wind turbine, the GE1.5sle power curve [39] is used to calculate the power output P_{out} at wind speed v:

$$P_{out}(v) = \begin{cases} 0, & v < 2.0 \\ 0.3\,v^3, & 2.0 \le v < 12.8 \\ 629.1, & 12.8 \le v \le 18.0 \\ 0, & v > 18.0 \end{cases} \tag{3}$$

where v is expressed in m/s and P_{out} in kW.

2) 3D wake effect model: For wind farm layout optimization problems, the wake effect is the primary factor affecting the wind speed v of downstream turbines. Currently, several 2D wake models, such as the Gaussian model and the Jensen model [6], [40], are widely employed. However, 2D models capture wake interactions only in the horizontal plane. In practical wind farms with complex terrain, turbine layouts are optimized in three dimensions. Consequently, compared with 2D wake models, as presented in Fig. 3, 3D wake models additionally account for vertical wind shear and wake diffusion in both the vertical and horizontal directions, providing a more realistic spatial distribution of wind speed. In this study, the 3D Jensen-Gaussian wake model is utilized. The method for computing the wind speed of each turbine [41], [42], [43] is described in the supplementary material.

B. 3D Wind Farm Noise Model

In this study, we adopt the noise model proposed by Hubbard et al. [44], which is based on aerodynamic theory and has been validated by the National Aeronautics and Space Administration. This model decomposes wind turbine noise into three main sources: inflow turbulence noise, trailing-edge interaction noise, and trailing-edge separation noise. Accordingly, the second objective, f_2, is defined as the average overall A-weighted sound level across multiple predefined noise-monitoring sites. The detailed calculation procedures [44], [45], [46] are provided in the supplementary material.

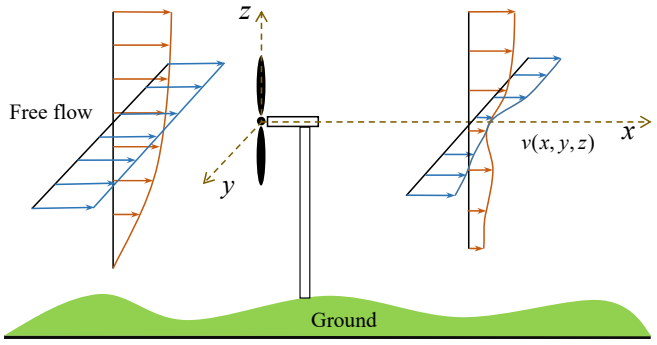


Fig. 3. Schematic diagram of the 3D wake effect model.

C. 3D Wind Farm Cost Constraint Model

Economic viability is a critical consideration in the CMWFLOP. To ensure the economic feasibility of the optimized layouts, the total project cost must not exceed a prescribed budget ε. In this study, two cost components are considered: (1) the cost of the turbines and (2) the cost associated with land use. The specific calculation methods [12], [13], [47], [48] are provided in the supplementary material.

IV. Proposed Framework

In this section, we present the proposed framework, including the main architecture, the pre-training stage, the fine-tuning strategy for the Transformer autoencoder, and the learning-assisted evolutionary optimization.

A. Main Architecture

In order to provide a comprehensive overview of the proposed TAAE, Fig. 4 summarizes the overall workflow, and **Algorithm** 1 presents the corresponding pseudocode. Specifically, a collection of candidate wind farm layouts is first used to pre-train the Transformer autoencoder, allowing the model to learn the spatial characteristics of the turbine placement. Subsequently, an initial solution set is randomly generated within the high-dimensional binary search space. Each individual is then integer-encoded, with each dimension storing the index of a candidate turbine location. At each generation, the Transformer autoencoder is fine-tuned using the current layouts together with their relative fitness values. Offspring are then generated in the latent space via DE, and layouts containing conflicts are repaired using a Gaussian-based resampling strategy. Environmental selection is performed based on the CDP and NSGA-II. This process is repeated until the termination criterion is met. In this way, the trained Transformer autoencoder acts as a learning model that can effectively guide the constrained multiobjective evolutionary search.

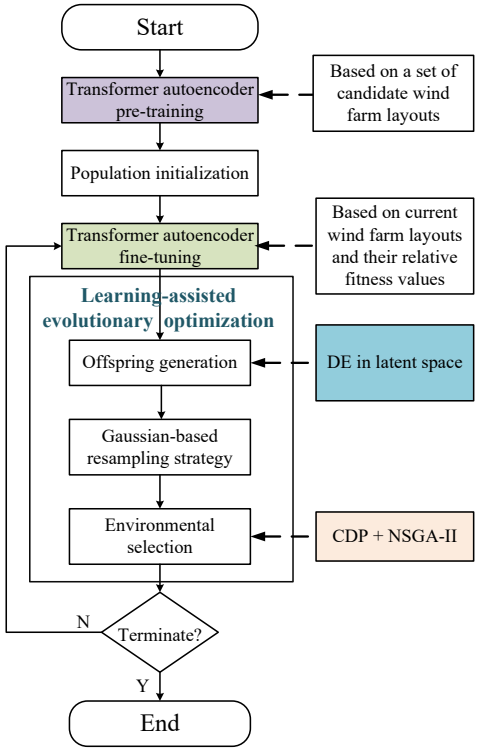


Fig. 4. The workflow of TAAE.

Algorithm 1: Main framework of TAAE

Input: Population size N; maximum number of fitness evaluations *maxFEs*
Output: Final population $\mathcal{P}$

```
Pre-train the Transformer autoencoder model;
Initialize the population P with N individuals;
FEs ← N;
while FEs ≤ maxFEs do
    Fine-tune the Transformer autoencoder model;
    /*Learning-assisted evolutionary optimization*/
    (P, FEs) ← EvolutionaryOptimization(P,
      Transformer autoencoder model, FEs);
end
```

B. Pre-Training Stage

In this study, as illustrated in Fig. 5, to provide strong reconstruction capability for wind farm layouts, we design a Transformer autoencoder neural network and pre-train it on a collection of candidate layouts before the evolutionary optimization stage.

Specifically, each candidate layout is encoded as a high-dimensional sparse binary vector $\vec{x}_b$. For example,

$$\vec{x}_b = (0,1,0,0,1,0,1,0,0,1,0,1)\,. \tag{4}$$

Here, 1 indicates that a wind turbine is installed at the corresponding grid cell, whereas 0 indicates otherwise. The dimension of $\vec{x}_b$ is N_g, where N_g denotes the total number of candidate grid cells covering the wind farm area. Subsequently, the high-dimensional binary vector $\vec{x}_b$ is converted into an

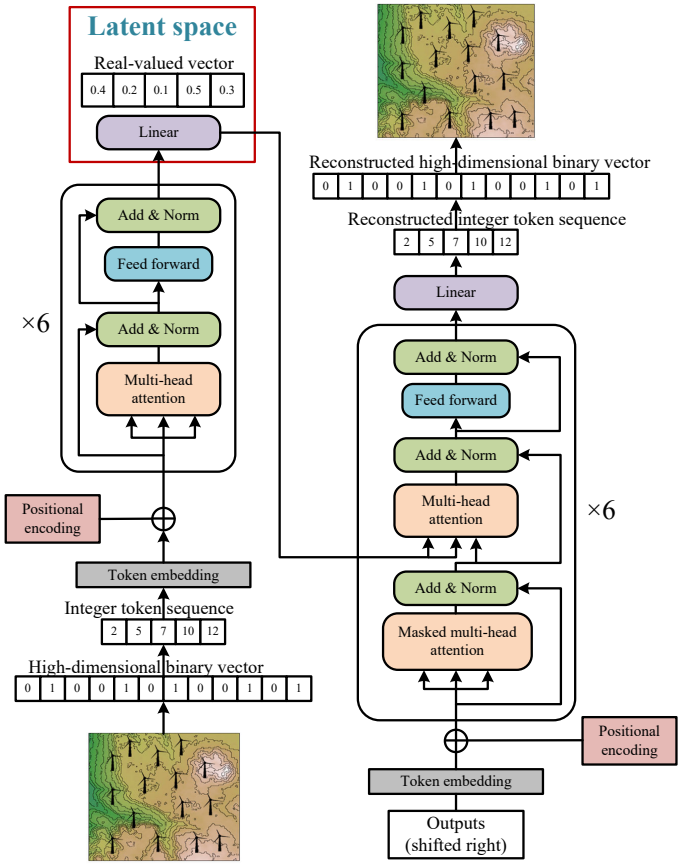


Fig. 5. Architecture of the pre-trained Transformer autoencoder.

integer token sequence $\vec{x}_{in}$. Specifically, the indices of the selected grid cells are represented as

$$\vec{x}_{in} = (i_1, i_2, \ldots, i_{N_w}), \tag{5}$$

where

$$\vec{x}_b(i_w) = 1, \quad w = 1, 2, \ldots, N_w. \tag{6}$$

Here, each element in $\vec{x}_{in}$ denotes the index of a selected candidate turbine location, and N_w denotes the number of wind turbines. For the above example, the corresponding integer token sequence is

$$\vec{x}_{in} = (2, 5, 7, 10, 12). \tag{7}$$

This integer token sequence then serves as the input for the Transformer autoencoder model.

The model structure and the pre-training process are described as follows:

Encoding: The input integer sequence is first mapped into token embeddings, which are then combined with positional embeddings and fed into a stack of Transformer encoder layers. The encoder stack consists of six Transformer encoder layers, each with a four-head self-attention module followed by a feed-forward sublayer, enabling the network to model spatial correlations across the entire wind farm layout. The resulting encoded representation is then compressed into a low-dimensional latent vector through a fully connected layer. This vector is denoted as $\vec{h}$, which serves as a compact and informative representation of the entire layout.

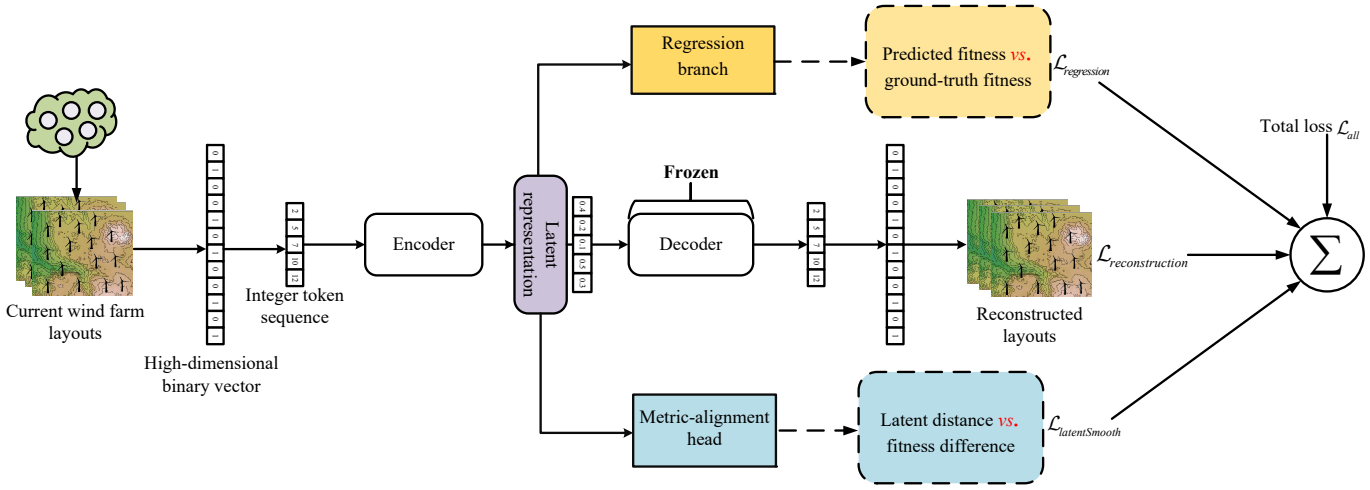


Fig. 6. Fine-tuning strategy for the Transformer autoencoder.

Decoding: During the decoding process, the compressed latent vector $\vec{h}$ is first linearly projected and reshaped into a sequence of embeddings, which serves as the static memory for the Transformer decoder. Subsequently, the token sequence is autoregressively reconstructed using a stack of six Transformer decoder layers and a right-shifted decoder input sequence. In addition, a special start-of-sequence token and a causal attention mask are employed to prevent information leakage. Afterwards, the decoder output is projected onto the vocabulary space to generate the predicted integer token sequence $\hat{\vec{x}}_{in}$. Finally, the reconstructed binary wind farm layout $\hat{\vec{x}}_b$ is obtained from $\hat{\vec{x}}_{in}$.

The goal of pre-training is to equip the Transformer autoencoder model with a strong reconstruction capability. To this end, a token-level cross-entropy loss between the input sequence and its reconstruction is minimized to encourage high similarity between the input and output. Consequently, the model can learn an effective encoder-decoder mapping and achieve strong reconstruction performance.

In this study, 100,000 wind farm layouts were randomly generated from the binary search space within the predefined wind farm area. Each generated layout contains exactly N_w turbines. These layouts were then used to pre-train the Transformer autoencoder, enabling it to learn spatial structural information from large-scale binary wind farm layouts and extract informative latent representations. The model was optimized using the Adam optimizer with a learning rate of 1×10^{-3} for 500 epochs with a batch size of 64. Both the token-embedding dimension and the latent bottleneck dimension of the Transformer autoencoder are set to 64. To verify the effectiveness of the pre-trained Transformer autoencoder model, it was evaluated on the test set. The experimental results showed that the pre-trained Transformer autoencoder model achieved a reconstruction accuracy of 0.99. The pre-trained weights can then be used to initialize the network for the subsequent learning-assisted evolutionary optimization stage.

C. Fine-Tuning Strategy for Transformer Autoencoder

To maintain reconstruction accuracy while improving the smoothness of the latent space, the pre-trained Transformer autoencoder is fine-tuned using the current population at each evolutionary generation. The specific fine-tuning strategy is presented in Fig. 6. At each evolutionary generation, each wind farm layout in the current population is associated with multiple objective values, namely energy output and noise impact, as well as a degree of constraint violation corresponding to the economic budget constraint. To incorporate this multiobjective and constraint information into the Transformer autoencoder, we employ the fitness-assignment method of SPEA2 combined with CDP. This procedure can assign a relative fitness value for each wind farm layout by considering its objective values and the degree of constraint violation.

To preserve reconstruction accuracy, the decoder parameters of the pre-trained Transformer autoencoder are frozen during fine-tuning. This helps maintain the reconstruction capability learned during pre-training. In addition, a reconstruction loss is introduced for the current population, where each individual represents a specific wind farm layout. The reconstruction loss is defined as follows:

$$\mathcal{L}_{reconstruction} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{N_w} \sum_{w=1}^{N_w} \ell_{CE} \left(\hat{y}_{i,w}, y_{i,w} \right), \tag{8}$$

where $\mathcal{B}$ denotes a mini-batch sampled from the current population, and $|\mathcal{B}|$ represents the batch size. N_w denotes the number of wind turbines, which is also the length of the input token sequence. In addition, $\hat{y}_{i,w}$ is the predicted token distribution for individual i at position w, $y_{i,w}$ is the corresponding ground-truth token label, and ℓ_{CE} denotes the token-level cross-entropy loss function.

Furthermore, to promote a continuous and smooth latent manifold, two lightweight auxiliary heads are added to the Transformer autoencoder: 1) a regression branch that predicts relative fitness values to provide pointwise supervision in the latent space, and 2) a metric-alignment head that regularizes the relative geometry of the latent space by encouraging

pairwise latent distances to reflect differences in relative fitness. These auxiliary heads are trained jointly with the reconstruction objective using a combined loss consisting of $\mathcal{L}_{reconstruction}$, $\mathcal{L}_{regression}$, and $\mathcal{L}_{latentSmooth}$.

1) **Regression branch:** We attach a lightweight regression branch to the Transformer autoencoder to predict the relative fitness value of each individual, thereby strengthening the alignment between latent representations and fitness values. Before being fed into the regression branch, each latent vector is normalized using L_2 normalization. The regression branch consists of a hidden layer followed by a linear output layer. Given the normalized latent vector $\bar{\vec{h}}_i$ of sample i, the hidden transformation is presented as follows:

$$\vec{z}_i = \sigma(W_1 \bar{\vec{h}}_i + \vec{b}_1), \tag{9}$$

where W_1 and $\vec{b}_1$ are the weight matrix and bias vector of the hidden layer, respectively, $\sigma(\cdot)$ denotes the activation function, which is implemented as ReLU in this study, and $\vec{z}_i$ is the hidden layer output. The prediction $\hat{s}_i$ is then produced by a linear transformation:

$$\hat{s}_i = \vec{w}_2^{\top} \vec{z}_i + b_2, \tag{10}$$

where $\vec{w}_2$ and b_2 are the weight vector and bias term of the linear output layer, respectively.

For the regression branch, the loss function $\mathcal{L}_{regression}$ is defined as follows:

$$\mathcal{L}_{regression} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(\hat{s}_i - \tilde{r}_i\right)^2, \tag{11}$$

where $\tilde{r}_i$ denotes the normalized relative fitness of sample i, obtained by applying min–max normalization to the relative fitness calculated using SPEA2 with CDP.

2) **Metric-alignment head:** To make the latent space fitness-aware, we introduce a metric-alignment head that encourages consistency between pairwise latent distances and the corresponding differences in relative fitness values. Specifically, given a mini-batch, the latent smoothness loss $\mathcal{L}_{latentSmooth}$ is formulated as follows:

$$\mathcal{L}_{latentSmooth} = \frac{1}{|\mathcal{B}|} \sum_{(i,j) \in Pair} \left(\|\bar{\vec{h}}_i - \bar{\vec{h}}_j\|_2 - |\tilde{r}_i - \tilde{r}_j| \right)^2, \tag{12}$$

where *Pair* is a collection of $|\mathcal{B}|$ randomly sampled index pairs within the batch, $\|\bar{\vec{h}}_i - \bar{\vec{h}}_j\|_2$ denotes the Euclidean distance between the normalized latent representations of samples i and j, and $|\tilde{r}_i - \tilde{r}_j|$ is the absolute difference between their normalized relative fitness values. In this way, layouts with similar fitness values are pulled closer in the latent space, whereas layouts with larger fitness differences are pushed farther apart.

By combining these three losses, the total loss $\mathcal{L}_{all}$ is defined as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{reconstruction} + \lambda_{reg}\, \mathcal{L}_{regression} + \lambda_{smooth}\, \mathcal{L}_{latentSmooth}, \tag{13}$$

where λ_{reg} and λ_{smooth} are nonnegative trade-off hyperparameters for the fitness regression loss and latent smoothness loss, respectively. In this study, λ_{reg} and λ_{smooth} are set to 30 and 1, respectively. By minimizing the loss $\mathcal{L}_{all}$ via backpropagation, the Transformer autoencoder learns to compress large-scale binary layouts into continuous and smooth latent representations.

To evaluate the effectiveness of the proposed fine-tuning strategy, a set of 3,000 candidate wind farm layouts was randomly generated. For simplicity, the relative fitness values of the wind farm layouts were independently sampled from the interval [0,1] and used to fine-tune the pre-trained Transformer autoencoder. To clearly visualize the latent-space geometry, we projected the latent vectors of the 3,000 wind farm layouts into two dimensions using t-SNE [49]. The points were color-coded according to relative fitness, with darker colors indicating lower fitness.

Fig. 7 shows the latent-space distributions before and after fine-tuning. As shown in Fig. 7(a), before fine-tuning, the latent representations were diffusely distributed and showed little organization with respect to relative fitness. In contrast, as shown in Fig. 7(b), after fine-tuning, the latent space became more structured, with smooth fitness gradients emerging across the manifold. This result suggests that fine-tuning generates a more fitness-aligned latent space, which can facilitate subsequent learning-assisted constrained multiobjective optimization.

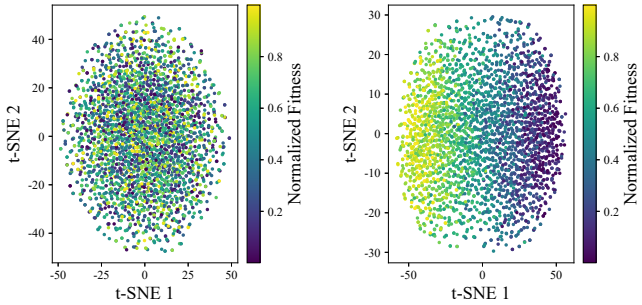


Fig. 7. Visualization of the latent-space distribution. (a) Before fine-tuning. (b) After fine-tuning.

D. Learning-Assisted Evolutionary Optimization

During the evolutionary process, the Transformer autoencoder is fine-tuned at each generation and then employed to assist constrained multiobjective evolutionary optimization. **Algorithm** 2 provides the pseudocode of this procedure, which primarily consists of two components: (1) Offspring generation and (2) Offspring repair.

Offspring generation: As shown in line 2 of **Algorithm** 2, the current population $\mathcal{P}$ is encoded into the latent space, yielding the latent population $\mathcal{H}_p \in \mathbb{R}^{N \times D_L}$, where N is the population size and D_L is the latent-space dimension. Subsequently, guided by CDP and NSGA-II [24], tournament selection is performed on $\mathcal{H}_p$ to construct two parent latent matrices, denoted as $\mathcal{H}_1 \in \mathbb{R}^{N \times D_L}$ and $\mathcal{H}_2 \in \mathbb{R}^{N \times D_L}$. Next, the DE operator is applied in the latent space to generate a latent mutant matrix $\mathcal{H}_m$ [50]:

$$\mathcal{H}_m = \mathcal{H}_p + F_{DE}(\mathcal{H}_1 - \mathcal{H}_2), \tag{14}$$

Algorithm 2: EvolutionaryOptimization

Input: $\mathcal{P}$, Transformer autoencoder model, *FEs*
Output: $\mathcal{P}$, *FEs*

/*Latent-space offspring generation*/
$\mathcal{H}_p \leftarrow$ Transformer autoencoder model.encode($\mathcal{P}$);
$\mathcal{H}_o \leftarrow$ Generate N latent offspring based on $\mathcal{H}_p$ using the DE operator;
$\mathcal{O} \leftarrow$ Transformer autoencoder model.decode($\mathcal{H}_o$);
/*Offspring repair*/
$\mathcal{O} \leftarrow$ Remove duplicate solutions from $\mathcal{O}$ and discard offspring identical to individuals in $\mathcal{P}$;
$\mathcal{O} \leftarrow$ Repair $\mathcal{O}$ using the Gaussian-based resampling strategy;
Evaluate $\mathcal{O}$ on all objectives and constraints;
$\mathcal{C} \leftarrow \mathcal{O} \cup \mathcal{P}$;
$\mathcal{P} \leftarrow$ Select N individuals from $\mathcal{C}$ based on CDP and NSGA-II;
Update *FEs*;

where F_{DE} is the scaling factor, fixed at 0.3, which controls the degree of mutation. Subsequently, crossover is performed between $\mathcal{H}_p$ and $\mathcal{H}_m$. For each latent offspring, one randomly selected latent dimension is inherited from the mutant latent vector, while the remaining dimensions are retained from the corresponding current latent vector. Afterwards, polynomial mutation [51] is applied to each latent dimension, with a mutation rate of $1/D_L$ and a distribution index of 20. Through the above process, the latent offspring $\mathcal{H}_o$ are generated. Finally, $\mathcal{H}_o$ is decoded back into the original integer space by the Transformer autoencoder model, yielding the offspring population $\mathcal{O}$.

Offspring repair: First, to reduce redundant evaluations and preserve population diversity, duplicate solutions are removed from the offspring population $\mathcal{O}$, and any offspring identical to individuals in the current population $\mathcal{P}$ are discarded.

Subsequently, for wind farm layout optimization, each grid cell can host at most one turbine, and turbines must not overlap with noise-monitoring sites. Therefore, a Gaussian-based resampling strategy is proposed to repair layouts containing conflicts. Specifically, conflicts in the generated layouts are first detected, including duplicate turbine placements and overlaps with noise-monitoring sites. When the number of conflicting turbines is not greater than two, a Gaussian distribution is fitted to the coordinates of the nonconflicting turbines. The mean of the distribution captures the layout center, while the covariance reflects the spatial spread of the turbine locations. Let $\vec{x}_i$ denote the coordinate vector of the i-th nonconflicting turbine, and let q be the number of nonconflicting turbines. The mean $\hat{\mu}$ and covariance $\hat{\Sigma}$ are computed as follows:

$$\hat{\mu} = \frac{1}{q}\sum_{i=1}^{q}\vec{x}_i, \tag{15}$$

$$\hat{\Sigma} = \frac{1}{q-1}\sum_{i=1}^{q}\left(\vec{x}_i - \hat{\mu}\right)\left(\vec{x}_i - \hat{\mu}\right)^{\top}. \tag{16}$$

The conflicting turbines are then resampled from the Gaussian distribution $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$, and the sampled coordinates are projected onto the nearest admissible unoccupied grid cells. This procedure preserves the spatial structure of the current layout and can help reduce the land-use cost. By contrast, when more than two conflicting turbines are detected, the entire wind farm layout is randomly reinitialized.

After the offspring have been repaired, as shown in lines 8–11 of **Algorithm** 2, the resulting offspring $\mathcal{O}$ are evaluated. Subsequently, the parents $\mathcal{P}$ and offspring $\mathcal{O}$ are merged, and environmental selection is carried out based on CDP and NSGA-II to form a new population.

V. Experimental Setting

This section presents a clear description of the wind farm layout optimization problems, the compared algorithms, the performance metrics, and the parameter settings.

A. Wind Farm Layout Optimization Problems and Competitors

To thoroughly evaluate the performance of the proposed TAAE algorithm, we constructed a comprehensive set of test problems that incorporates (1) eight distinct wind scenarios, denoted as WS1-WS8, (2) two different configurations of noise-monitoring sites, denoted as NC1 and NC2, and (3) three different economic budget constraints. Specifically, the eight wind scenarios capture a range of wind conditions typically encountered at real-world wind farm sites, including both unidirectional and complex multidirectional flows, as well as uniform and variable wind speed distributions. The wind rose figures for these scenarios, which illustrate the joint distributions of wind direction and speed, are presented in Fig. 8. In addition, as illustrated in Fig. 9, two different noise-monitoring site configurations are constructed. Moreover, three distinct economic budget thresholds are imposed to constrain the total layout cost, set to 0.6×10^6, 0.8×10^6, and 1.0×10^6, respectively. Based on differences in noise-monitoring configurations and budget constraints, these test problems are grouped into six test suites, denoted as C1-C6. All optimization problems are constructed based on the real terrain data of the Zhangbei wind farm area. The area is discretized into a 20×20 grid, and our final goal is to determine the locations of 15 wind turbines.

In this study, seven state-of-the-art CMOEAs are introduced to evaluate the effectiveness of the proposed algorithm. These compared algorithms include MTCMO [52], CMOES [53], CSEMT [54], DRLOS-EMCMO [55], IMTCMO [56], MCCMO [57], and DVCEA [58]. Specifically, MTCMO constructs a dynamic auxiliary evolutionary subtask to assist the main constrained multiobjective optimization task. CMOES performs an even search within the promising regions of CMOPs by integrating an enhanced ϵ-constraint handling method. CSEMT adopts a multitasking paradigm, in which dynamically evolving auxiliary tasks are employed to explore promising regions and provide valuable knowledge for the main task. DRLOS-EMCMO leverages deep Q-learning to adaptively select suitable evolutionary operators based on the current evolutionary state. IMTCMO adopts a cooperative

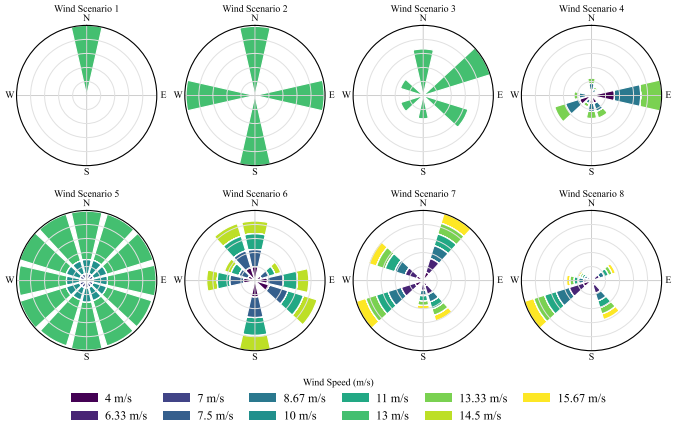
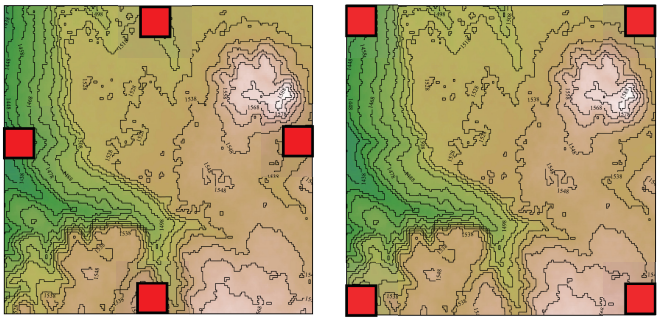


Fig. 8. Wind rose figures for eight different wind scenarios.



(a) (b)

Fig. 9. Two configurations of noise-monitoring sites. Red squares denote the monitoring sites. (a) Noise-monitoring configuration 1. (b) Noise-monitoring configuration 2.

coevolutionary framework consisting of a main population and an auxiliary population, and enhances its search capability through newly designed mutation operators, thereby effectively balancing convergence and diversity. MCCMO addresses CMOPs through the collaboration of multiple populations, which are activated or merged under appropriate conditions to enhance optimization performance. Regarding DVCEA, it separately optimizes constraint-related and constraint-independent decision variables in order to balance feasibility and diversity during the evolutionary process.

B. Performance Metrics and Parameter Settings

To evaluate the performance of the proposed algorithm, three widely used performance metrics are adopted: hypervolume (HV) [59], inverted generational distance (IGD) [60], and inverted generational distance plus (IGD^+) [61]. HV measures the volume of the objective space region bounded by a predefined reference point and the obtained nondominated solution set. Thus, a larger HV value indicates better algorithmic performance. IGD calculates the average distance from a set of uniformly distributed reference points sampled from the true Pareto front to the obtained solution set. Accordingly, a smaller IGD value indicates better performance. For real-world problems in which the true Pareto front is unknown, the reference Pareto front is estimated as the union of nondominated solutions obtained by all algorithms across all runs. IGD^+ is a modified version of IGD whose comparison results are always consistent with Pareto dominance. Similarly, a smaller IGD^+ value indicates better performance. For each wind farm layout optimization problem, both the proposed algorithm and all baseline algorithms were independently run 16 times. The means and standard deviations of the experimental results are reported in the tables. Statistical comparisons among algorithms were conducted using the Wilcoxon rank-sum test with a significance level of 0.05 [62]. The symbols "+", "-", and "≈" denote that the performance of the competing algorithm is significantly better, significantly worse, or statistically comparable to that of TAAE, respectively. Moreover, the Friedman test combined with Bonferroni–Dunn post-hoc analysis was implemented by using the KEEL software [63] to compare the HV, IGD, and IGD^+ values. This test assigns a ranking score to each algorithm, with lower scores representing better performance. Furthermore, the maximum number of fitness evaluations is limited to 10,000, and the population size is fixed at 100 for all algorithms. All remaining parameters of the compared algorithms are set according to the recommendations in their respective original papers. With respect to the proposed algorithm, the number of fine-tuning epochs is set to 10, and the dimension of the latent space is set to 64.

The experimental environment is as follows: MATLAB 2024a, Intel(R) Core i9-13900K CPU, and 128 GB of RAM. For the deep learning component of the proposed algorithm, the Transformer autoencoder model of TAAE was implemented, trained, and evaluated using Python, with computations performed on an NVIDIA RTX 4090 GPU with 24 GB of VRAM. All experiments were conducted using the PlatEMO platform [64].

VI. Experimental Results and Analysis

In this section, the proposed TAAE algorithm and the competing algorithms are evaluated on six test suites. Subsequently, the effectiveness of the learning-assisted evolutionary search is analyzed. In addition, the effectiveness of the pretraining stage and fine-tuning strategy is discussed. Afterwards, the reconstruction accuracy of the Transformer autoencoder during the evolutionary process is visualized. Finally, parameter sensitivity analysis experiments are conducted.

A. Comparison Results Across Different Test Instances

1) Experiments on eight wind scenarios with noise-monitoring site configuration 1 and economic constraint 1 (C1): Table I, Table S1[1], and Table S2 report the mean values and standard deviations of HV, IGD, and IGD^+ achieved by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C1 test suite. Overall, TAAE performs best under most wind scenarios. For example, in terms of the HV metric, TAAE consistently attains higher values than all competing algorithms across

[1]Due to space limitations, Tables S1-S21 and Figs. S1-S51 are provided in the supplementary material.

TABLE I
STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C1 TEST SUITE. THE GRAY BACKGROUND HIGHLIGHTS THE BEST RESULT FOR EACH TEST INSTANCE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	1.0131e-1 (1.60e-3) —	1.0109e-1 (1.78e-3) —	1.0229e-1 (1.20e-3) —	9.9077e-2 (1.64e-3) —	9.8566e-2 (1.50e-3) —	1.0288e-1 (1.81e-3) —	1.0117e-1 (1.73e-3) —	1.0497e-1 (1.66e-3)
WS2	7.7828e-2 (1.14e-3) —	7.7545e-2 (1.19e-3) —	7.8199e-2 (1.30e-3) —	7.5363e-2 (1.11e-3) —	7.6477e-2 (6.90e-4) —	7.7871e-2 (1.32e-3) —	7.7327e-2 (8.55e-4) —	8.1066e-2 (1.04e-3)
WS3	3.6694e-2 (6.91e-4) —	3.6332e-2 (4.44e-4) —	3.6363e-2 (5.65e-4) —	3.5285e-2 (4.69e-4) —	3.5861e-2 (3.76e-4) —	3.6634e-2 (5.18e-4) —	3.6344e-2 (5.09e-4) —	3.7994e-2 (3.04e-4)
WS4	3.2649e-2 (4.85e-4) —	3.2341e-2 (4.99e-4) —	3.2286e-2 (4.46e-4) —	3.1323e-2 (5.38e-4) —	3.1783e-2 (3.48e-4) —	3.2239e-2 (5.03e-4) —	3.2356e-2 (3.71e-4) —	3.3284e-2 (4.72e-4)
WS5	3.3423e-2 (4.81e-4) —	3.3576e-2 (6.57e-4) —	3.3585e-2 (6.89e-4) —	3.2365e-2 (3.84e-4) —	3.3066e-2 (3.55e-4) —	3.3704e-2 (4.91e-4) —	3.3351e-2 (5.41e-4) —	3.4989e-2 (3.86e-4)
WS6	2.8411e-2 (5.15e-4) —	2.8351e-2 (4.67e-4) —	2.8325e-2 (4.44e-4) —	2.7452e-2 (3.50e-4) —	2.7989e-2 (2.10e-4) —	2.8274e-2 (4.14e-4) —	2.8028e-2 (4.49e-4) —	2.9362e-2 (5.17e-4)
WS7	1.5584e-2 (2.81e-4) —	1.5473e-2 (3.20e-4) —	1.5277e-2 (3.69e-4) —	1.4971e-2 (2.45e-4) —	1.5203e-2 (2.42e-4) —	1.5486e-2 (3.46e-4) —	1.5532e-2 (2.10e-4) —	1.6161e-2 (3.47e-4)
WS8	4.8529e-2 (5.73e-4) —	4.8682e-2 (5.78e-4) —	4.8781e-2 (7.64e-4) —	4.7300e-2 (5.66e-4) —	4.8415e-2 (4.15e-4) —	4.8540e-2 (5.70e-4) —	4.8632e-2 (5.09e-4) —	4.9669e-2 (5.76e-4)
+/ − / ≈	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

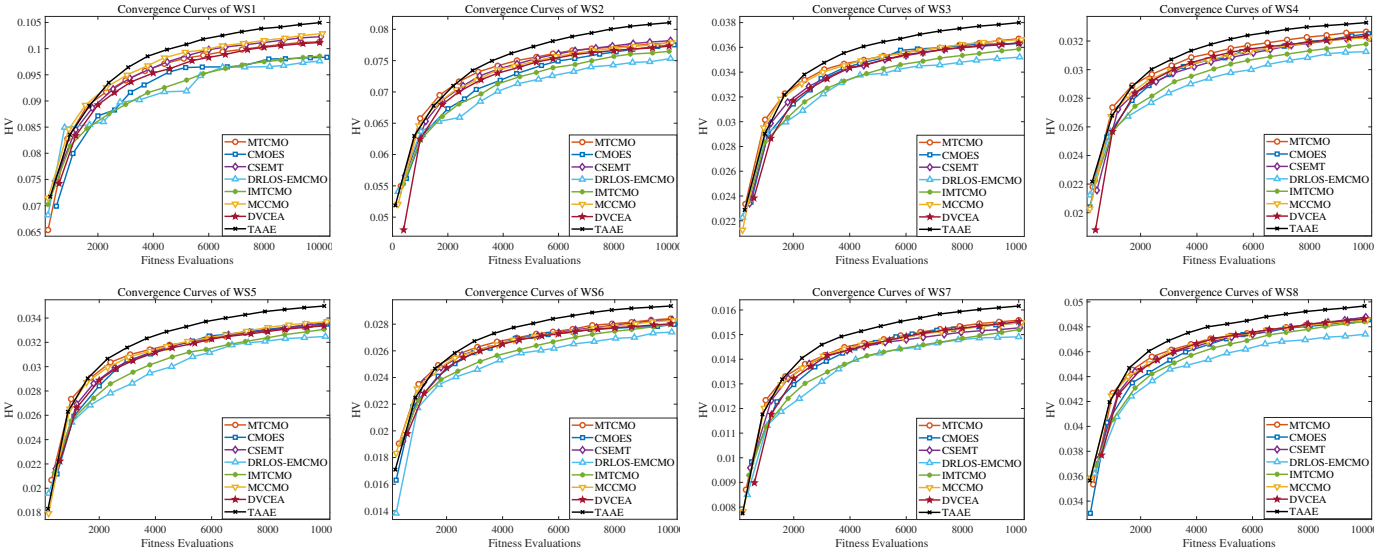


Fig. 10. HV convergence curves on the C1 test suite under eight different wind scenarios.

all wind scenarios. The HV convergence curves for the C1 suite are shown in Fig. 10, where TAAE converges fastest and achieves the best performance. In addition, to further demonstrate the effectiveness of TAAE, the Friedman test was conducted based on the experimental results of HV, IGD, and IGD$^+$. As depicted in Fig. 11, TAAE ranks first with respect to all three metrics.

Furthermore, to present a clear overview of the experimental results under eight different wind scenarios, the final solution sets with the median HV value among 16 runs obtained by TAAE and other compared algorithms are presented in Fig. 12, Fig. 13, and Figs. S1–S6, respectively. The experimental results demonstrate that, compared with other algorithms, the proposed TAAE can obtain feasible, well-distributed, and well-converged solution sets. The reasons are as follows: Before the evolutionary search, a Transformer autoencoder is pre-trained on numerous wind farm layouts to ensure strong reconstruction performance. During the evolutionary process, the fitness information of the wind farm layouts is incorporated into the fine-tuning process of the Transformer autoencoder. This enables high-dimensional binary layouts to be encoded into a low-dimensional, real-valued, continuous, and smooth latent manifold, thereby facilitating subsequent constrained multiobjective evolutionary optimization.

2) Experiments on eight wind scenarios with noise-monitoring site configuration 1 and economic constraint 2

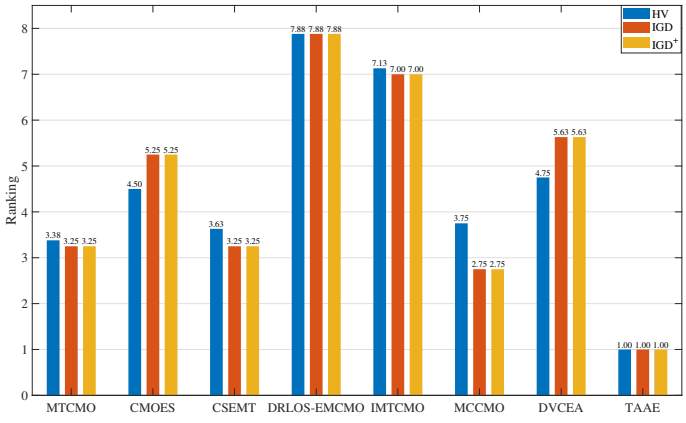


Fig. 11. Friedman test results for TAAE and other compared algorithms on the C1 test suite.

(C2): Table II, Table S3, and Table S4 summarize the mean values and standard deviations of the HV, IGD, and IGD$^+$ obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C2 test suite. The experimental results demonstrate that TAAE achieves overall superior performance compared with the other competitors in terms of the three metrics. In addition, based on the obtained experimental results, the Friedman test was conducted. As shown in Fig. S7, TAAE achieves the best

TABLE II
STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C2 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	9.9013e-2 (1.74e-3) —	9.7330e-2 (1.27e-3) —	9.9702e-2 (1.53e-3) —	9.5603e-2 (1.24e-3) —	9.6539e-2 (1.59e-3) —	1.0013e-1 (1.51e-3) —	9.7667e-2 (1.77e-3) —	1.0115e-1 (7.61e-4)
WS2	7.6557e-2 (1.42e-3) —	7.5981e-2 (1.20e-3) —	7.6418e-2 (1.01e-3) —	7.3796e-2 (9.80e-4) —	7.5038e-2 (8.54e-4) —	7.6630e-2 (1.23e-3) —	7.6248e-2 (1.08e-3) —	7.9094e-2 (1.51e-3)
WS3	3.1265e-2 (6.91e-4) —	3.0921e-2 (5.85e-4) —	3.0977e-2 (5.67e-4) —	3.0046e-2 (5.88e-4) —	3.0504e-2 (4.38e-4) —	3.0747e-2 (5.61e-4) —	3.0855e-2 (4.98e-4) —	3.2009e-2 (7.83e-4)
WS4	3.4624e-2 (4.73e-4) —	3.4519e-2 (3.94e-4) —	3.4501e-2 (5.69e-4) —	3.3711e-2 (6.89e-4) —	3.4171e-2 (3.18e-4) —	3.4603e-2 (5.22e-4) —	3.4838e-2 (4.49e-4) —	3.5466e-2 (6.21e-4)
WS5	4.5741e-2 (1.12e-3) —	4.6051e-2 (9.24e-4) —	4.6444e-2 (8.87e-4) —	4.4402e-2 (8.45e-4) —	4.5386e-2 (5.15e-4) —	4.6052e-2 (1.22e-3) —	4.6126e-2 (7.20e-4) —	4.7413e-2 (8.94e-4)
WS6	2.4142e-2 (3.66e-4) —	2.4058e-2 (4.02e-4) —	2.4021e-2 (4.21e-4) —	2.3256e-2 (5.47e-4) —	2.3697e-2 (4.53e-4) —	2.4013e-2 (5.69e-4) —	2.4185e-2 (4.50e-4) —	2.5088e-2 (3.91e-4)
WS7	1.4818e-2 (2.89e-4) —	1.4712e-2 (2.36e-4) —	1.4686e-2 (3.33e-4) —	1.4258e-2 (2.27e-4) —	1.4471e-2 (1.59e-4) —	1.4764e-2 (3.69e-4) —	1.4942e-2 (2.39e-4) ≈	1.5190e-2 (4.43e-4)
WS8	2.8860e-2 (5.33e-4) —	2.8660e-2 (4.56e-4) —	2.8614e-2 (4.97e-4) —	2.7970e-2 (2.82e-4) —	2.8485e-2 (3.68e-4) —	2.8921e-2 (4.30e-4) —	2.8745e-2 (3.51e-4) —	2.9938e-2 (3.74e-4)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/7/1	

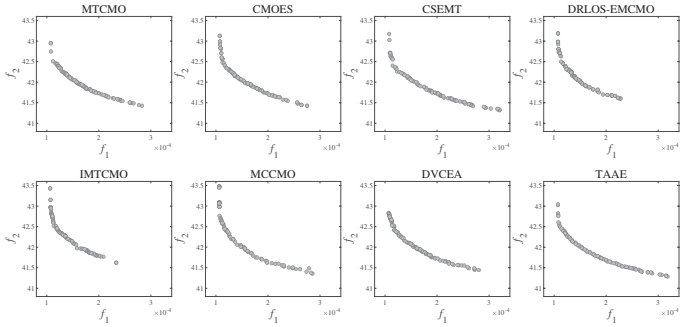


Fig. 12. Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS1.

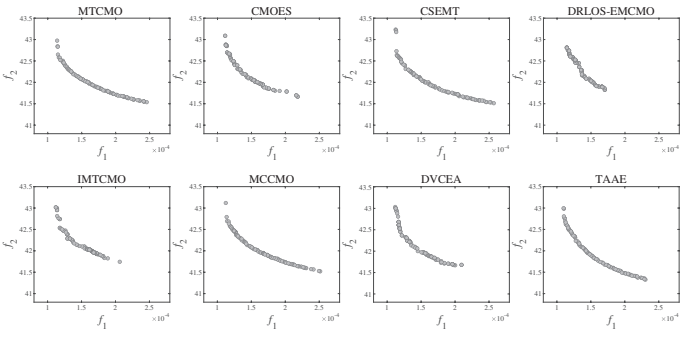


Fig. 13. Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS2.

overall ranking. Furthermore, the final solution sets obtained by TAAE and other compared algorithms on the C2 test suite under eight wind scenarios are shown in Figs. S8–S15. These figures further demonstrate that TAAE produces feasible, well-converged and well-distributed solution sets compared with the baseline algorithms.

3) Experiments on eight wind scenarios with noise-monitoring site configuration 1 and economic constraint 3 (C3): Table S5, Table S6, and Table S7 present the mean values and standard deviations of the HV, IGD, and IGD$^+$ achieved by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C3 test suite. As shown in these tables, TAAE achieves the best overall performance. In addition, as displayed in Fig. S16, the Friedman test results indicate that TAAE achieves the first rank. Moreover, the final solution sets obtained by TAAE and other peer algorithms are visualized in Figs. S17–S24, indicating that the proposed TAAE can provide feasible, well-converged, and well-distributed solution sets under most wind scenarios.

The experimental results of TAAE and the other seven competitors on the C4, C5, and C6 test suites are provided in the supplementary material.

B. Effectiveness of the Learning-Assisted Evolutionary Search

In this subsection, to evaluate the effectiveness of the proposed learning-assisted evolutionary search, two variants of TAAE are introduced for comparison:

- TAAE_Binary: A binary encoding scheme is used, where 0 indicates that no turbine is placed at a candidate site, whereas 1 indicates that a turbine is placed.
- TAAE_Integer: An integer encoding method is employed, where each dimension stores the index of a selected candidate site.

In contrast, the original TAAE framework transforms the high-dimensional binary optimization space into a low-dimensional, real-valued, continuous, and smooth manifold to facilitate evolutionary optimization. To compare their performance, TAAE_Binary, TAAE_Integer, and the original TAAE are tested on the C1 test suite. The experimental results are summarized in Table S17. As shown in the table, the original TAAE consistently achieves the best overall performance. The performance advantages can be explained as follows. Compared with TAAE_Binary, the original TAAE significantly reduces the dimensionality of the decision space and learns a smoother and more continuous latent representation. This enables the evolutionary algorithm to explore the search space more efficiently and identify feasible, well-converged, and well-distributed solution sets. Meanwhile, compared with TAAE_Integer, although the latent representation in TAAE may be higher-dimensional than the compact integer representation, it preserves the continuity and smoothness of the latent space. This facilitates gradient-free evolutionary search and leads to satisfactory performance. In summary, the original TAAE framework achieves superior performance by leveraging the low-dimensional, real-valued, continuous, and smooth latent space.

C. Effectiveness of the Pre-Training Stage and Fine-Tuning Strategy

A key component of this work is the design of the pre-

training stage and fine-tuning strategy for the Transformer autoencoder. To assess the effectiveness of these components, an ablation study is conducted in this subsection. Specifically, two variants are designed:

- TAAE_1: The pre-training stage of TAAE is omitted.
- TAAE_2: The fine-tuning strategy of TAAE is removed.

Then, TAAE_1, TAAE_2, and the original TAAE are evaluated on the C1 test suite under eight different wind scenarios. Table S18 presents the HV results obtained by TAAE_1, TAAE_2, and the original TAAE. It can be observed that the original TAAE consistently outperforms TAAE_1 and TAAE_2 across all wind scenarios on the C1 test suite. The reasons are as follows. For TAAE_1, omitting the pre-training stage eliminates the strong reconstruction prior learned by the Transformer autoencoder. As a result, fine-tuning the Transformer autoencoder model solely based on the current population during the evolutionary process significantly reduces reconstruction accuracy, which leads to degraded overall performance. For TAAE_2, the absence of the fine-tuning strategy prevents the model from effectively leveraging fitness information from the current wind farm layouts. Consequently, the latent space cannot be further shaped into a fitness-aligned and smooth manifold, which impairs the effectiveness of evolutionary search and results in inferior performance. In summary, these results indicate that the best performance is achieved through the joint use of the proposed pre-training stage and fine-tuning strategy within the TAAE framework.

D. Visualization of Reconstruction Accuracy During the Evolutionary Processes

During the learning-assisted evolutionary optimization process, the Transformer autoencoder encodes the original large-scale binary wind farm layouts into low-dimensional, real-valued, continuous, and smooth latent representations. Subsequently, after offspring are generated in this latent space by an evolutionary search algorithm, the decoder maps them back to binary layouts. This process poses a significant challenge to the reconstruction capability of the Transformer autoencoder. To verify the reconstruction capability of the Transformer autoencoder during the evolutionary process, we directly apply the proposed TAAE to solve the C1 test suite under WS1.

To quantitatively assess reconstruction accuracy, we employ two complementary metrics: element level accuracy and sequence level accuracy. The element level metric measures the proportion of correctly reconstructed turbine positions within each layout. In contrast, the sequence level metric assesses the proportion of layouts in which the entire binary sequence is perfectly reconstructed. Clearly, the sequence level metric imposes a stricter criterion than the element level metric.

The reconstruction accuracies based on both metrics are illustrated in Fig. 14. As shown in the figure, both the element level and sequence level accuracies exceed 0.9, thereby demonstrating the strong reconstruction capability of the Transformer autoencoder during the evolutionary optimization process.

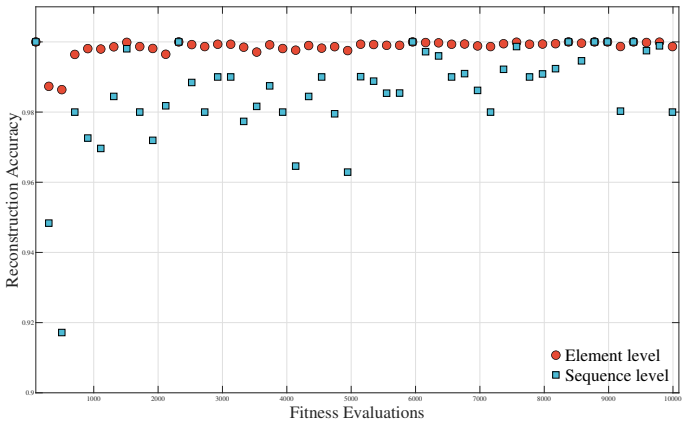


Fig. 14. Element level and sequence level reconstruction accuracies on the C1 test suite under WS1.

E. Parameter Sensitivity Analysis

First, we conducted a parameter sensitivity analysis of the scaling factor F_{DE} in the DE search algorithm, where F_{DE} controls the degree of mutation. In this subsection, five variants of TAAE with $F_{DE} = 0.1$, 0.3, 0.5, 0.7, and 0.9 were evaluated on the C1 test suite. The experimental results are summarized in Table S19. It can be observed that the performance of TAAE is not sensitive to the value of F_{DE}. Based on these results, the parameter F_{DE} is set to 0.3 in this study.

Second, at each evolutionary generation, the number of fine-tuning epochs can affect the runtime of the proposed algorithm. Let *Epoch* denote the number of fine-tuning epochs. We conducted experiments with *Epoch* set to 5, 10, and 15, and the results are reported in Table S20. The results show that *Epoch* $= 10$ outperforms *Epoch* $= 5$ on one test instance and achieves performance comparable to that of *Epoch* $= 15$. In addition, *Epoch* $= 10$ requires less computational time than *Epoch* $= 15$. Therefore, *Epoch* is set to 10 in this study.

Finally, since the large-scale binary search space is transformed into a low-dimensional, real-valued, continuous, and smooth latent search space in this study, we conducted an ablation study to examine the impact of the latent-space dimension D_L on optimization performance. Specifically, D_L was set to 32, 64, and 70, and the three variants were evaluated on the C1 test suite under the eight wind scenarios. The experimental results are summarized in Table S21, demonstrating that $D_L = 64$ yields the best performance. The reasons can be analyzed as follows. When $D_L = 32$, the latent dimension is too small, which may restrict the reconstruction capability of the model and lead to poor optimization performance. In contrast, when $D_L = 70$, the latent dimension becomes too large, which may cause redundancy. Moreover, since our framework employs a basic DE operator, higher-dimensional latent spaces may require more advanced search operators to be effective. Therefore, $D_L = 64$ achieves the best performance and is adopted in this study.

VII. Conclusion

In this paper, we propose the TAAE framework for solving complex CMWFLOPs. By carefully considering real-world

factors, the WFLOP is formulated as a large-scale binary CMOP that integrates a 3D wake model, with the aim of simultaneously maximizing power output and minimizing noise impact while satisfying an economic budget constraint. The TAAE framework is designed to map the large-scale binary layouts onto a low-dimensional, real-valued, continuous, and smooth latent manifold. Specifically, during the pre-training stage, a collection of candidate wind farm layouts is fed into the Transformer autoencoder, thereby encouraging the latent space to capture the spatial features of the layouts. Subsequently, during the evolutionary search process, a regression branch and a metric-alignment head are added to the Transformer autoencoder model. The regression branch is used to predict the relative fitness values, and the metric-alignment head is used to encourage consistency between pairwise latent-space distances and the corresponding fitness differences. The current layouts, along with their relative fitness values, are fed into the Transformer autoencoder for fine-tuning, encouraging the latent space to form a continuous and smooth manifold. Afterwards, learning-assisted constrained multiobjective optimization is performed on this manifold, during which the Gaussian-based resampling strategy is utilized to repair conflicting layouts. Furthermore, CDP, NSGA-II, and DE are integrated into the proposed framework for constraint handling, environmental selection, and latent-space offspring generation, respectively. The proposed algorithm is evaluated on six test suites, each comprising eight wind scenarios. Compared with other state-of-the-art algorithms, the proposed framework demonstrates superior performance.

In future work, we will employ time-series neural networks to directly predict wind farm power generation [65]. In addition, we will apply the proposed TAAE framework to other large-scale binary optimization problems to further examine its generalizability and applicability.

REFERENCES

[1] L. Liu, G. He, M. Wu, G. Liu, H. Zhang, Y. Chen, J. Shen, and S. Li, "Climate change impacts on planned supply–demand match in global wind and solar energy systems," *Nature Energy*, vol. 8, no. 8, pp. 870–880, Aug. 2023.

[2] Y. Wang, R. Wang, K. Tanaka, P. Ciais, J. Penuelas, Y. Balkanski, J. Sardans, D. Hauglustaine, W. Liu, X. Xing, J. Li, S. Xu, Y. Xiong, R. Yang, J. Cao, J. Chen, L. Wang, X. Tang, and R. Zhang, "Accelerating the energy transition towards photovoltaic and wind in China," *Nature*, vol. 619, no. 7971, pp. 761–767, Jul. 2023.

[3] G. M. Joselin Herbert, S. Iniyan, E. Sreevalsan, and S. Rajapandian, "A review of wind energy technologies," *Renewable and Sustainable Energy Reviews*, vol. 11, no. 6, pp. 1117–1145, Aug. 2007.

[4] S. Gao, K. Wang, S. Tao, T. Jin, H. Dai, and J. Cheng, "A state-of-the-art differential evolution algorithm for parameter estimation of solar photovoltaic models," *Energy Conversion and Management*, vol. 230, p. 113784, Feb. 2021.

[5] M. S. Chowdhury, K. S. Rahman, V. Selvanathan, N. Nuthammachot, M. Suklueng, A. Mostafaeipour, A. Habib, Md. Akhtaruzzaman, N. Amin, and K. Techato, "Current trends and prospects of tidal energy technology," *Environment, Development and Sustainability*, vol. 23, no. 6, pp. 8179–8194, Jun. 2021.

[6] Z. Lei, S. Gao, Z. Zhang, H. Yang, and H. Li, "A chaotic local search-based particle swarm optimizer for large-scale complex wind farm layout optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1168–1180, May 2023.

[7] W. Zhang, H. Sun, J. Gao, G. Yan, and M. Yang, "A two-stage ultra-short-term wind power forecasting method based on transitional weather identification and meteorological prediction error propagation," *IEEE Transactions on Sustainable Energy*, vol. 17, no. 1, pp. 295-306, Jan. 2026.

[8] J. Li, Z. Zhang, T. Zheng, J. Tang, Z. Lei, and S. Gao, "Discrete bi-population differential evolution for optimizing complex wind farm layouts in diverse terrains," *Energy*, vol. 335, p. 137885, Oct. 2025.

[9] J. Yang, Y. Song, W. Ding, J. Tang, Z. Lei, and S. Gao, "Multi-attention-powered learning genetic algorithm for real-world 3D wind farm layout optimization," *Swarm and Evolutionary Computation*, vol. 97, p. 102018, Aug. 2025.

[10] Z. Lei, S. Gao, Y. Wang, Y. Yu, and L. Guo, "An adaptive replacement strategy-incorporated particle swarm optimizer for wind farm layout optimization," *Energy Conversion and Management*, vol. 269, p. 116174, Oct. 2022.

[11] Z. Zhang, J. Li, Z. Lei, Q. Zhu, J. Cheng, and S. Gao, "Reinforcement learning-based particle swarm optimization for wind farm layout problems," *Energy*, vol. 313, p. 134050, Dec. 2024.

[12] S. R. Moreno, J. Pierezan, L. D. S. Coelho, and V. C. Mariani, "Multi-objective lightning search algorithm applied to wind farm layout optimization," *Energy*, vol. 216, p. 119214, Feb. 2021.

[13] C. Zhang, H. Li, X. Li, J. Cheng, Z. Lei, and S. Gao, "Probabilistic bootstrap-based evolutionary algorithm for three-objective wind farm turbine position optimization," *Swarm and Evolutionary Computation*, vol. 96, p. 101972, Jul. 2025.

[14] K. Yang, M. Zhang, S. Yang, Y. Song, X. Dong, Y. Deng, and X. Deng, "Pareto frontier for multi-objective wind farm layout optimization balancing power production and turbine fatigue life," *Renewable Energy*, vol. 252, p. 123429, Oct. 2025.

[15] X. Zhong, X. Yao, K. Qiao, D. Gong, and Y. Jin, "An indicator-based evolutionary algorithm for large-scale constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 1, pp. 271–285, Feb. 2026.

[16] L. Si, X. Zhang, Y. Zhang, S. Yang, and Y. Tian, "An efficient sampling approach to offspring generation for evolutionary large-scale constrained multiobjective optimization," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 9, no. 3, pp. 2080–2092, Jun. 2025.

[17] A. Pan, H. Liu, Y. Shan, and B. Shen, "A bilevel coevolution framework with knowledge transfer for large-scale optimization and its application in multiperiod economic dispatch," *Engineering Applications of Artificial Intelligence*, vol. 141, p. 109775, Feb. 2025.

[18] J. Liu, Y. Wang, P.-Q. Huang, and S. Jiang, "CaR: A cutting and repulsion-based evolutionary framework for mixed-integer programming problems," *IEEE Transactions on Cybernetics*, vol. 52, no. 12, pp. 13129–13141, Dec. 2022.

[19] S. Liu, J. Li, Q. Lin, Y. Tian, J. Li, and K. C. Tan, "Evolutionary large-scale multiobjective optimization via autoencoder-based problem transformation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 4, pp. 2709–2722, Aug. 2024.

[20] G. Yuan, B. Wang, B. Xue, and M. Zhang, "Particle swarm optimization for efficiently evolving deep convolutional neural networks using an autoencoder-based encoding strategy," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 5, pp. 1190–1204, Oct. 2024.

[21] S. Tan, Y. Wang, G. Sun, and T. Pang, "A surrogate-assisted high-dimensional mixed-variable evolutionary framework and its application to vehicle lightweighting design," *IEEE Transactions on Evolutionary Computation*, vol. 30, no. 2, pp. 836–850, Apr. 2026.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[23] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evolutionary Methods for Design, Optimization, and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[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, Apr. 2002.

[25] Y. Yu, V. O. K. Li, J. C. K. Lam, K. Chan, and Q. Zhang, "CTDI: CNN-Transformer-based spatial-temporal missing air pollution data imputation," *IEEE Transactions on Big Data*, vol. 11, no. 5, pp. 2443–2456, Oct. 2025.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," in *Proc. North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Jun. 2019, pp. 4171–4186.

[27] L. Floridi and M. Chiriatti, "GPT-3: Its nature, scope, limits, and consequences," *Minds and Machines*, vol. 30, no. 4, pp. 681–694, Dec. 2020.

[28] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[29] Z.-H. Zhan, J.-Y. Li, S. Kwong, and J. Zhang, "Learning-aided evolution for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 6, pp. 1794–1808, Dec. 2023.

[30] Q. Dang, G. Zhang, L. Wang, Y. Yu, S. Yang, and X. He, "Transformer-based intelligent prediction model for multimodal multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 20, no. 1, pp. 34–49, Feb. 2025.

[31] H. Kamyshanska and R. Memisevic, "The potential energy of an autoencoder," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 6, pp. 1261–1273, Jun. 2015.

[32] Y. Ma, B. Shen, A. Pan, and J. Xue, "Constraint landscape knowledge assisted constrained multiobjective optimization," *Swarm and Evolutionary Computation*, vol. 90, p. 101685, Oct. 2024.

[33] Y. Ma, B. Shen, and A. Pan, "Constrained evolutionary optimization based on dynamic knowledge transfer," *Expert Systems with Applications*, vol. 240, p. 122450, Apr. 2024.

[34] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. D. Goodman, "Push and pull search for solving constrained multiobjective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, Feb. 2019.

[35] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K. C. Tan, and Y. Jin, "Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 52, no. 9, pp. 9559–9572, Sep. 2022.

[36] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multiobjective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, Feb. 2021.

[37] J. Liang, K. Qiao, K. Yu, B. Qu, C. Yue, W. Guo, and L. Wang, "Utilizing the relationship between unconstrained and constrained Pareto fronts for constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 53, no. 6, pp. 3873–3886, Jun. 2023.

[38] K. Qiao, J. Liang, Z. Liu, K. Yu, C. Yue, and B. Qu, "Evolutionary multitasking with global and local auxiliary tasks for constrained multiobjective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 10, pp. 1951–1964, Oct. 2023.

[39] F. Bai, X. Ju, S. Wang, W. Zhou, and F. Liu, "Wind farm layout optimization using adaptive evolutionary algorithm with Monte Carlo Tree Search reinforcement learning," *Energy Conversion and Management*, vol. 252, p. 115047, Jan. 2022.

[40] S. Tao, S. Kuenzel, Q. Xu, and Z. Chen, "Optimal micro-siting of wind turbines in an offshore wind farm using Frandsen–Gaussian wake model," *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 4944–4954, Nov. 2019.

[41] H. Kim, C. Singh, and A. Sprintson, "Simulation and estimation of reliability in a wind farm considering the wake effect," *IEEE Transactions on Sustainable Energy*, vol. 3, no. 2, pp. 274–282, Apr. 2012.

[42] X. Gao, B. Li, T. Wang, H. Sun, H. Yang, Y. Li, Y. Wang, and F. Zhao, "Investigation and validation of 3D wake model for horizontal-axis wind turbines based on filed measurements," *Applied Energy*, vol. 260, p. 114272, Feb. 2020.

[43] W. Liu, X. Zhu, K. Wang, X. Gao, S. Zhang, L. Dong, Z. Shi, H. Lu, and J. Zhou, "Derivation and verification of Gaussian terrain wake model based on wind field experiment," *Processes*, vol. 10, no. 12, p. 2731, Dec. 2022.

[44] H. H. Hubbard and K. P. Shepherd, "Aeroacoustics of large wind turbines," *The Journal of the Acoustical Society of America*, vol. 89, no. 6, pp. 2495–2508, 1991.

[45] S. Sukaj, G. Iannace, G. Ciaburro, and F. Iannace, "Wind turbines noise: Predictions and measurements," *Romanian Journal of Acoustics and Vibration*, vol. 18, no. 2, pp. 69–75, 2021.

[46] X. Wu, W. Hu, Q. Huang, C. Chen, M. Z. Jacobson, and Z. Chen, "Optimizing the layout of onshore wind farms to minimize noise," *Applied Energy*, vol. 267, p. 114896, Jun. 2020.

[47] G. Mosetti, C. Poloni, and B. Diviacco, "Optimization of wind turbine positioning in large windfarms by means of a genetic algorithm," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 51, no. 1, pp. 105–116, Jan. 1994.

[48] R. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, vol. 1, pp. 132–133, 1972.

[49] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008.

[50] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[51] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[52] K. Qiao et al., "Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 642–656, Jun. 2023.

[53] F. Ming, W. Gong, and Y. Jin, "Even search in a promising region for constrained multiobjective optimization," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 474–486, Feb. 2024.

[54] K. Qiao, J. Liang, K. Yu, X. Ban, C. Yue, B. Qu, and P. N. Suganthan, "Constraints separation based evolutionary multitasking for constrained multi-objective optimization problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 8, pp. 1819–1835, Aug. 2024.

[55] F. Ming, W. Gong, L. Wang, and Y. Jin, "Constrained multiobjective optimization with deep reinforcement learning assisted operator selection," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 4, pp. 919–931, Apr. 2024.

[56] K. Qiao, J. Liang, K. Yu, C. Yue, H. Lin, D. Zhang, and B. Qu, "Evolutionary constrained multiobjective optimization: Scalable high-dimensional constraint benchmarks and algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 4, pp. 965–979, Aug. 2024.

[57] J. Zou, R. Sun, Y. Liu, Y. Hu, S. Yang, J. Zheng, and K. Li, "A multi-population evolutionary algorithm using new cooperative mechanism for solving multiobjective problems with multiconstraint," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 267–280, Feb. 2024.

[58] X. Ban, J. Liang, K. Qiao, K. Yu, Y. Wang, J. Peng, and B. Qu, "A decision variables classification-based evolutionary algorithm for constrained multi-objective optimization problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 12, no. 9, pp. 1830-1849, Sep. 2025.

[59] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[60] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[61] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *Proc. Evolutionary Multi-Criterion Optimization*, Mar. 2015, pp. 110–125.

[62] P. B. Dao, "On Wilcoxon rank sum test for condition monitoring and fault detection of wind turbines," *Applied Energy*, vol. 318, p. 119209, Jul. 2022.

[63] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. Del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Computing*, vol. 13, no. 3, pp. 307–318, Feb. 2009.

[64] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multiobjective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, Nov. 2017.

[65] C.-L. Huang, Y.-K. Wu, Q.-T. Phan, C.-C. Tsai, and J.-S. Hong, "Enhancing wind power forecasts via bias correction technologies for numerical weather prediction model," *IEEE Transactions on Industry Applications*, vol. 61, no. 4, pp. 5406–5419, Jul.–Aug. 2025.

Yuhang Ma received the B.Eng. degree in communication engineering from Donghua University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree with the School of Information and Intelligent Science, Donghua University, Shanghai, China. His current research interests include evolutionary optimization, computational intelligence, and machine learning. He is currently a very active Reviewer for many international journals.

Shangce Gao received his Ph.D. degree in Innovative Life Science from the University of Toyama, Toyama, Japan, in 2011. He is currently a Professor with the Faculty of Engineering at the University of Toyama. His research interests focus on brain-inspired neural networks and their applications in medical diagnosis and drug discovery. He serves as an Associate Editor for several international journals, including *IEEE Transactions on Neural Networks and Learning Systems*, and *IEEE/CAA Journal of Automatica Sinica*.

Anqi Pan received the Ph.D. degree in control science and engineering from Tongji University, Shanghai, China, in 2019. She is currently an Associate Professor with the School of Information and Intelligent Science, Donghua University, Shanghai, China. Her main research interests include computational intelligence and machine learning for multi-objective, robust, and constrained optimization, and their applications in process industry and intelligent manufacturing systems.

Bo Shen (Senior Member, IEEE) received the B.Sc. degree in mathematics from Northwestern Polytechnical University, Xi'an, China, in 2003, and the Ph.D. degree in control theory and control engineering from Donghua University, Shanghai, China, in 2011. He is currently a Professor with the School of Information and Intelligent Science, Donghua University. From 2009 to 2010, he was a Research Assistant with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, China. From 2010 to 2011, he was a visiting Ph.D. student with the Department of Information Systems and Computing, Brunel University London, Uxbridge, U.K. From 2011 to 2013, he was a Research Fellow (Scientific co-worker) with the Institute for Automatic Control and Complex Systems, University of Duisburg-Essen, Duisburg, Germany. He has published around 120 papers in refereed international journals. His research interests include nonlinear control and filtering, stochastic control and filtering, as well as complex networks and neural networks. Prof. Shen serves (or has served) as an Associate Editor or an Editorial Board Member for eight international journals, including *Systems Science and Control Engineering*, *the Journal of the Franklin Institute*, *the Asian Journal of Control*, *Circuits, Systems, and Signal Processing*, *Neurocomputing*, *Robotic Intelligence and Automation*, *Neural Processing Letters*, and *Mathematical Problems in Engineering*. He is a Program Committee Member for many international conferences.

Supplementary Material for "Transformer Autoencoder-Assisted Evolutionary Framework for Constrained Multiobjective 3D Wind Farm Layout Optimization"

Yuhang Ma, Shangce Gao, Anqi Pan, and Bo Shen

CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. 3D Wake Effect Model

The wake effect is a primary factor that affects the wind speed of downstream turbines. To account for wind shear, the vertical inflow velocity, $v(z)$, is described by the following power-law expression [41]:

$$v(z) = v_{ref}\left(\frac{z + z_{hub}}{z_{ref}}\right)^{\alpha}, \tag{1}$$

where z denotes the vertical displacement relative to the hub center, v_{ref} is the incoming wind speed at the reference height z_{ref}, z_{hub} denotes the turbine hub height, and α is the wind shear exponent, which is typically determined by local atmospheric conditions and terrain.

The axial induction factor a, derived from the thrust coefficient C_T, is computed as follows [42]:

$$a = \frac{1 - \sqrt{1 - C_T}}{2}. \tag{2}$$

The initial effective wake radius r_0 is calculated as follows:

$$r_0 = r_d\sqrt{\frac{1-a}{1-2a}}, \tag{3}$$

where r_d represents the rotor radius of the wind turbine.

The wake expansion coefficients k_z and k_y in the vertical and horizontal directions, respectively, are expressed as follows [43]:

$$\begin{aligned} k_z &= 0.243346\, C_T^{0.4297} I_0^{0.4707}, \\ k_y &= 0.18265\, C_T^{0.2566} I_0^{0.2808}, \end{aligned} \tag{4}$$

where I_0 is the turbulence intensity at the turbine inflow. The wake radii at the downstream position x in the vertical and horizontal directions, denoted by r_z and r_y, respectively, are defined as follows:

$$\begin{aligned} r_z &= k_z x + r_0, \\ r_y &= k_y x + r_0. \end{aligned} \tag{5}$$

The standard deviations of the Gaussian velocity-deficit distribution, σ_z and σ_y, are then determined as follows:

$$\begin{aligned} \sigma_z &= \frac{r_z}{2.58}, \\ \sigma_y &= \frac{r_y}{2.58}. \end{aligned} \tag{6}$$

Consequently, the wind speed $v\,(x, y, z)$ at an arbitrary spatial point can be formulated as follows:

$$\begin{aligned} v(x,y,z) = v_{ref}\Bigg[&\left(\frac{z + z_{hub}}{z_{ref}}\right)^{\alpha} - \frac{4ar_0^2}{\sigma_z\, r_z\, \sqrt{2\pi}} e^{-\frac{z^2}{2\sigma_z^2} - \frac{y^2}{2\sigma_y^2}} \\ &- \frac{a}{r_z}\, e^{-\frac{y^2}{2\sigma_y^2}} \Bigg\{ \frac{z_{hub}}{\alpha+1}\Bigg[\left(\frac{z_{hub} + r_0}{z_{hub}}\right)^{\alpha+1} \\ &- \left(\frac{z_{hub} - r_0}{z_{hub}}\right)^{\alpha+1}\Bigg] - 2\, r_0 \Bigg\}\Bigg]. \end{aligned} \tag{7}$$

To ensure physical realism, the computed wind speed is constrained to be nonnegative:

$$v(x, y, z) = \max(v(x, y, z), 0). \tag{8}$$

II. 3D Wind Farm Noise Model

Based on wind turbine acoustics, the noise model decomposes wind turbine noise into three principal sources.

1) Inflow turbulence noise: When incoming turbulent airflow strikes the rotor blades, inflow turbulence noise is generated. For the b-th one-third-octave band ($b = 1, \ldots, n_{band}$), the corresponding sound pressure level ($SPL_{1,b}$) is given by

$$SPL_{1,b} = 10\log_{10}\left[\frac{B\sin^2\varphi\,\rho^2\,C_{0.7}\,R\sigma^2 V_{0.7}^4}{r_1^2 c_0^2}\right] + K_a(f_b), \tag{9}$$

where B is the number of blades, $\sin^2\varphi$ is a directivity factor accounting for the angle φ between the rotor plane and the observer's line of sight, ρ denotes the air density, R is the rotor radius, $C_{0.7}$ is the blade chord length at $0.7R$, σ^2 is the mean square turbulent velocity of the incoming flow, $V_{0.7}$ is the blade forward speed at $0.7R$, r_1 denotes the reference distance from

the noise source to a reference observer, c_0 is the speed of sound in air, f_b represents the band center frequency, and $K_a(f_b)$ is a frequency-dependent correction term, measured in dB, applied to the inflow–noise spectrum.

2) Trailing-edge interaction noise: The noise generated by the interaction between the turbulent boundary layer and the blade trailing edge is quantified by the sound pressure level ($SPL_{2,b}$), which is calculated as follows:

$$SPL_{2,b} = 10\log_{10}\left\{\frac{V_r^5 B D_1 \delta l}{r_1^2}\left(\frac{S}{S_{max}}\right)^4\left[\left(\frac{S}{S_{max}}\right)^{1.5}+\frac{1}{2}\right]^{-4}\right\}+K_b, \tag{10}$$

where V_r is the local resultant flow velocity at the blade section, D_1 is the directivity factor for trailing-edge noise, δ is the boundary–layer thickness at the trailing edge, l is the spanwise length of the blade, the product δl denotes the effective radiating area, S is the Strouhal number for trailing–edge noise, $S_{\max}$ is the Strouhal number at which the trailing-edge noise reaches its maximum, and K_b, measured in dB, is an empirical scaling factor that determines the overall level of trailing-edge noise.

3) Trailing-edge separation noise: Owing to the sharp trailing-edge design of modern blades, the noise generated by flow separation is minor compared with the first two noise components and is therefore often neglected [44].

By considering the first two noise components and combining them in the energy domain, the band sound pressure level $L_{k,b}$ for turbine k in band b can be obtained as follows:

$$L_{k,b} = 10\log_{10}\left(10^{SPL_{1,b}/10}+10^{SPL_{2,b}/10}\right). \tag{11}$$

Furthermore, to account for the frequency-dependent sensitivity of human hearing, the A-weighted sound pressure level $L^{\mathrm{A}}_{k,b,p}$ from turbine k in band b at noise-monitoring site p can be calculated as follows:

$$L^{\mathrm{A}}_{k,b,p} = L_{k,b} - A_{k,p} + \Delta L_b, \tag{12}$$

where $A_{k,p}$ denotes the distance-dependent propagation attenuation from turbine k to noise-monitoring site p [45], and ΔL_b denotes the signed A-weighting correction applied to frequency band b [46].

Finally, the overall A-weighted sound level $L_{total,p}$ at noise-monitoring site p is obtained by energy summation over all N_w turbines and n_{band} frequency bands:

$$L_{total,p} = 10\log_{10}\left(\sum_{k=1}^{N_w}\sum_{b=1}^{n_{band}} 10^{L^{\mathrm{A}}_{k,b,p}/10}\right). \tag{13}$$

In this study, the noise impact of a wind farm layout under each wind scenario is evaluated using the average overall A-weighted sound level across all noise-monitoring sites.

III. 3D Wind Farm Cost Constraint Model

With respect to the cost constraint model, two cost components are considered: (1) turbine cost and (2) land-use cost. The calculation methods are described below.

1) Cost of Wind Turbines: Based on the empirical cost model proposed in [47], the aggregate turbine cost C_{turb} for N_w identical turbines can be approximated as

$$C_{turb} = N_w\left(\frac{2}{3}+\frac{1}{3}\,e^{-0.00174N_w^2}\right), \tag{14}$$

where N_w denotes the number of wind turbines.

2) Cost Associated with Land Use: The land-use cost C_{land} is assumed to be proportional to the occupied area and can be calculated as follows:

$$C_{land} = \xi \cdot S_\gamma, \tag{15}$$

where ξ is the unit land-cost coefficient and S_γ is the effective land area.

To estimate the land area occupied by the wind farm, two complementary area measures, $S_{\gamma 1}$ and $S_{\gamma 2}$, are first calculated and then averaged.

First, the convex hull of all turbine coordinates is computed using Graham's algorithm [48]. The area of the resulting polygon is then calculated using the shoelace formula [12]:

$$S_{\gamma 1} = \frac{1}{2}\left|\sum_{i=1}^{m-1} x_i y_{i+1} + x_m y_1 - \sum_{i=1}^{m-1} x_{i+1} y_i - x_1 y_m\right|, \tag{16}$$

where (x_i, y_i) are the Cartesian coordinates of the i–th hull vertex, and m is the number of vertices.

Second, to reduce the sensitivity of area estimation to irregular layouts [13], the rectangular area of the layout, $S_{\gamma 2}$, is computed as

$$S_{\gamma 2} = \eta\,\kappa, \tag{17}$$

where η and κ denote the width and length, respectively, of the minimum axis-aligned bounding rectangle enclosing all turbines in the horizontal plane.

The average of $S_{\gamma 1}$ and $S_{\gamma 2}$ is considered as the effective land area S_γ, which is given by

$$S_\gamma = \frac{1}{2}(S_{\gamma 1} + S_{\gamma 2}). \tag{18}$$

Thus, by considering both turbine cost and land-use cost, the total cost can be expressed as

$$C_{total} = C_{turb} + C_{land}. \tag{19}$$

Since the total cost C_{total} must not exceed the economic budget ε, the cost constraint is defined as

$$C_{total} \leq \varepsilon. \tag{20}$$

IV. Experimental Results and Analysis on the C4, C5, and C6 Test Suites

1) Experiments on eight wind scenarios with noise-monitoring site configuration 2 and economic constraint 1 (C4)

Table S8, Table S9, and Table S10 list the mean values and standard deviations of the HV, IGD, and IGD$^+$ obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C4 test suite. These tables indicate that the proposed TAAE achieves the best overall performance. In addition, the Friedman test was conducted based on the HV, IGD, and IGD$^+$ metrics, and the results are presented in Fig. S25. It can be observed that TAAE achieves the best rank, followed by CMOES. Furthermore, as shown in Figs. S26–S33, the final solution sets produced by TAAE under most wind scenarios are feasible, well-converged, and well-distributed.

2) Experiments on eight wind scenarios with noise-monitoring site configuration 2 and economic constraint 2 (C5)

Table S11, Table S12, and Table S13 provide the mean values and standard deviations of the HV, IGD, and IGD$^+$ obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C5 test suite. The results show that TAAE delivers the best overall performance. In addition, the Friedman test was conducted, and the results are presented in Fig. S34, which shows TAAE attains the best rank, followed by MTCMO. Moreover, the distributions of the final solution sets in Figs. S35–S42 confirm that TAAE can yield feasible, well-converged, and well-distributed solution sets.

3) Experiments on eight wind scenarios with noise-monitoring site configuration 2 and economic constraint 3 (C6)

Table S14, Table S15, and Table S16 present the mean values and standard deviations of the HV, IGD, and IGD$^+$ obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and the proposed TAAE on the C6 test suite. These results indicate that TAAE achieves superior performance compared with the baseline algorithms. In addition, the Friedman test was conducted based on the HV, IGD, and IGD$^+$ metrics, and the results are presented in Fig. S43. It can be observed that TAAE achieves the best rank. Furthermore, as presented in Figs. S44–S51, the final solution sets obtained by TAAE under most wind scenarios are feasible, well-converged, and well-distributed.

TABLE S1: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C1 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	6.6140e-2 (3.05e-2) —	7.1736e-2 (3.86e-2) —	4.2694e-2 (2.11e-2) —	1.1175e-1 (4.82e-2) —	1.3304e-1 (5.30e-2) —	3.6658e-2 (3.53e-2) —	6.9040e-2 (3.69e-2) —	1.6818e-2 (7.68e-3)
WS2	9.0191e-2 (4.25e-2) —	1.1479e-1 (4.66e-2) —	7.7285e-2 (4.47e-2) —	1.9238e-1 (5.36e-2) —	1.5994e-1 (3.30e-2) —	8.6763e-2 (4.88e-2) —	1.2561e-1 (4.53e-2) —	1.8519e-2 (1.46e-2)
WS3	1.0306e-1 (5.56e-2) —	1.6403e-1 (3.80e-2) —	1.0896e-1 (4.86e-2) —	1.9208e-1 (3.89e-2) —	1.7320e-1 (3.55e-2) —	9.8759e-2 (4.49e-2) —	1.4720e-1 (3.78e-2) —	2.7910e-2 (2.85e-2)
WS4	6.8585e-2 (3.55e-2) —	9.8354e-2 (4.12e-2) —	7.1399e-2 (3.78e-2) —	1.7312e-1 (4.48e-2) —	1.4109e-1 (3.46e-2) —	7.9288e-2 (5.02e-2) —	1.1576e-1 (3.91e-2) —	2.8379e-2 (3.57e-2)
WS5	9.1473e-2 (3.74e-2) —	9.2602e-2 (4.40e-2) —	7.2617e-2 (4.17e-2) —	1.4964e-1 (3.34e-2) —	1.2387e-1 (4.21e-2) —	5.8843e-2 (3.38e-2) —	9.8518e-2 (2.98e-2) —	1.7383e-2 (1.60e-2)
WS6	8.4903e-2 (5.02e-2) —	1.2558e-1 (5.13e-2) —	7.9322e-2 (4.64e-2) —	1.7560e-1 (5.14e-2) —	1.4620e-1 (4.35e-2) —	9.8332e-2 (3.96e-2) —	1.4545e-1 (4.78e-2) —	2.7022e-2 (3.43e-2)
WS7	7.1409e-2 (4.71e-2) —	1.2259e-1 (6.09e-2) —	1.2920e-1 (5.18e-2) —	1.7826e-1 (4.52e-2) —	1.4102e-1 (5.66e-2) —	1.0313e-1 (5.67e-2) —	1.1785e-1 (4.20e-2) —	2.9831e-2 (5.10e-2)
WS8	1.1682e-1 (3.41e-2) —	1.1895e-1 (3.32e-2) —	1.1405e-1 (4.86e-2) —	1.8171e-1 (3.96e-2) —	1.3650e-1 (2.03e-2) —	1.0742e-1 (3.89e-2) —	1.4477e-1 (3.41e-2) —	5.2271e-2 (2.30e-2)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

TABLE S2: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C1 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	5.9194e-2 (3.11e-2) —	6.4432e-2 (3.99e-2) —	3.4360e-2 (2.10e-2) —	1.0410e-1 (4.84e-2) —	1.2685e-1 (5.34e-2) —	2.8638e-2 (3.62e-2) —	6.2250e-2 (3.78e-2) —	9.1764e-3 (8.11e-3)
WS2	8.5238e-2 (4.33e-2) —	1.0967e-1 (4.76e-2) —	7.1594e-2 (4.59e-2) —	1.8680e-1 (5.45e-2) —	1.5537e-1 (3.38e-2) —	8.0700e-2 (5.00e-2) —	1.2014e-1 (4.64e-2) —	1.3088e-2 (1.54e-2)
WS3	9.5530e-2 (5.72e-2) —	1.5772e-1 (3.86e-2) —	1.0148e-1 (4.96e-2) —	1.8575e-1 (3.89e-2) —	1.6875e-1 (3.58e-2) —	9.1989e-2 (4.66e-2) —	1.4225e-1 (3.84e-2) —	2.0036e-2 (2.88e-2)
WS4	6.2207e-2 (3.76e-2) —	9.2739e-2 (4.25e-2) —	6.4661e-2 (3.91e-2) —	1.6811e-1 (4.67e-2) —	1.3610e-1 (3.54e-2) —	7.2570e-2 (5.14e-2) —	1.1050e-1 (4.00e-2) —	2.1451e-2 (3.68e-2)
WS5	8.3908e-2 (4.03e-2) —	8.6528e-2 (4.48e-2) —	6.5725e-2 (4.24e-2) —	1.4421e-1 (3.41e-2) —	1.1912e-1 (4.33e-2) —	5.1752e-2 (3.38e-2) —	9.2792e-2 (3.06e-2) —	1.0509e-2 (1.71e-2)
WS6	7.8933e-2 (5.17e-2) —	1.1982e-1 (5.31e-2) —	7.2446e-2 (4.70e-2) —	1.7032e-1 (5.24e-2) —	1.4183e-1 (4.45e-2) —	9.2135e-2 (4.09e-2) —	1.4021e-1 (4.94e-2) —	2.0058e-2 (3.46e-2)
WS7	6.6772e-2 (4.90e-2) —	1.2002e-1 (6.19e-2) —	1.2621e-1 (5.30e-2) —	1.7647e-1 (4.60e-2) —	1.3911e-1 (5.75e-2) —	1.0003e-1 (5.82e-2) —	1.1562e-1 (4.29e-2) —	2.6477e-2 (5.22e-2)
WS8	8.8907e-2 (3.51e-2) —	9.0773e-2 (3.34e-2) —	8.5480e-2 (4.89e-2) —	1.5533e-1 (4.03e-2) —	1.1310e-1 (1.97e-2) —	8.0717e-2 (3.91e-2) —	1.2185e-1 (3.57e-2) —	2.1013e-2 (2.32e-2)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

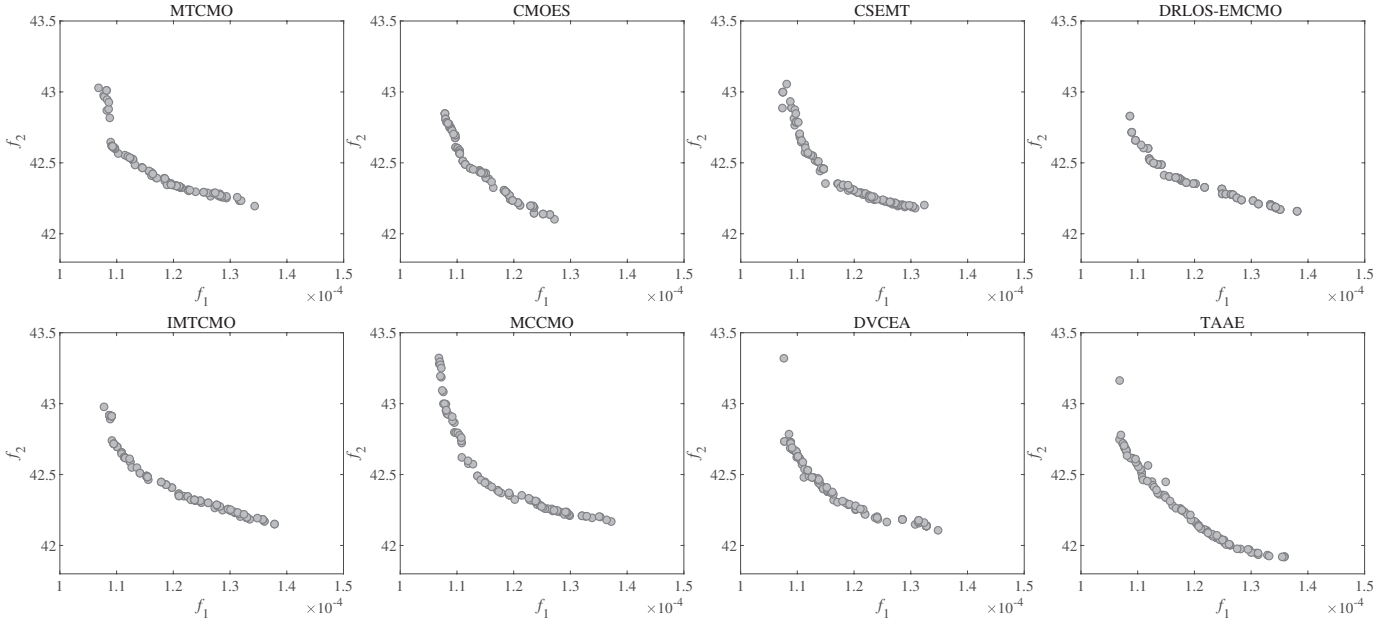


Fig. S1: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS3.

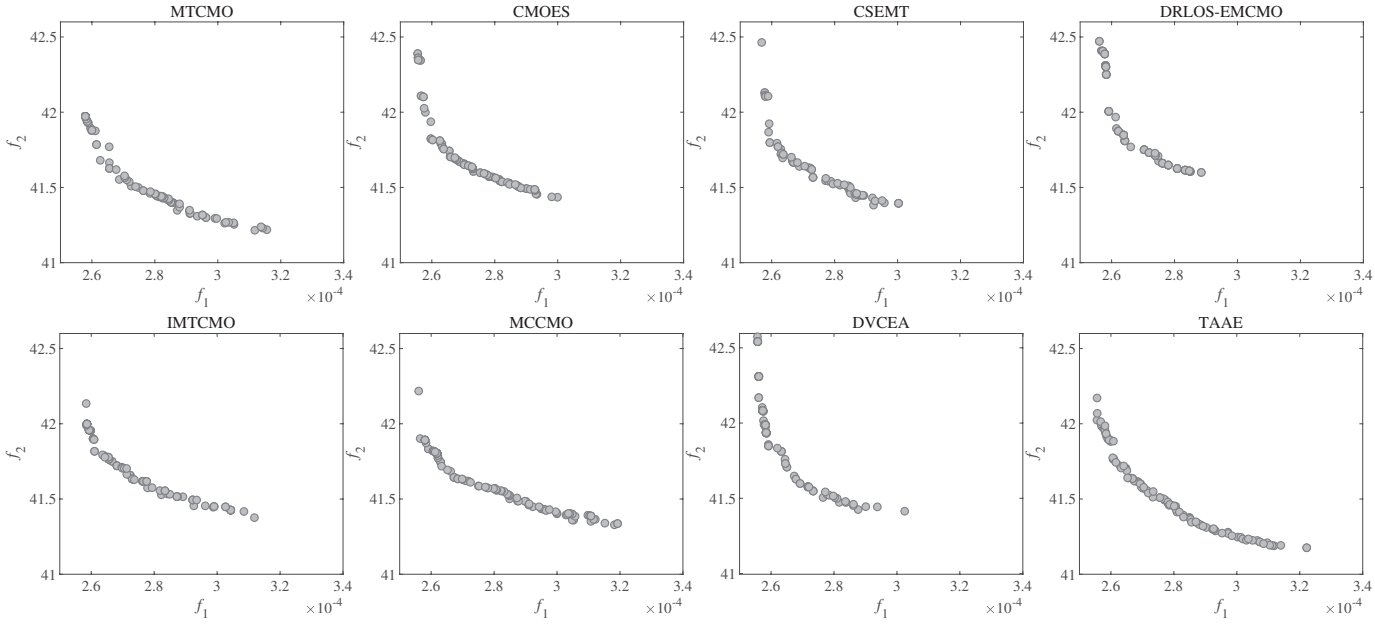


Fig. S2: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS4.

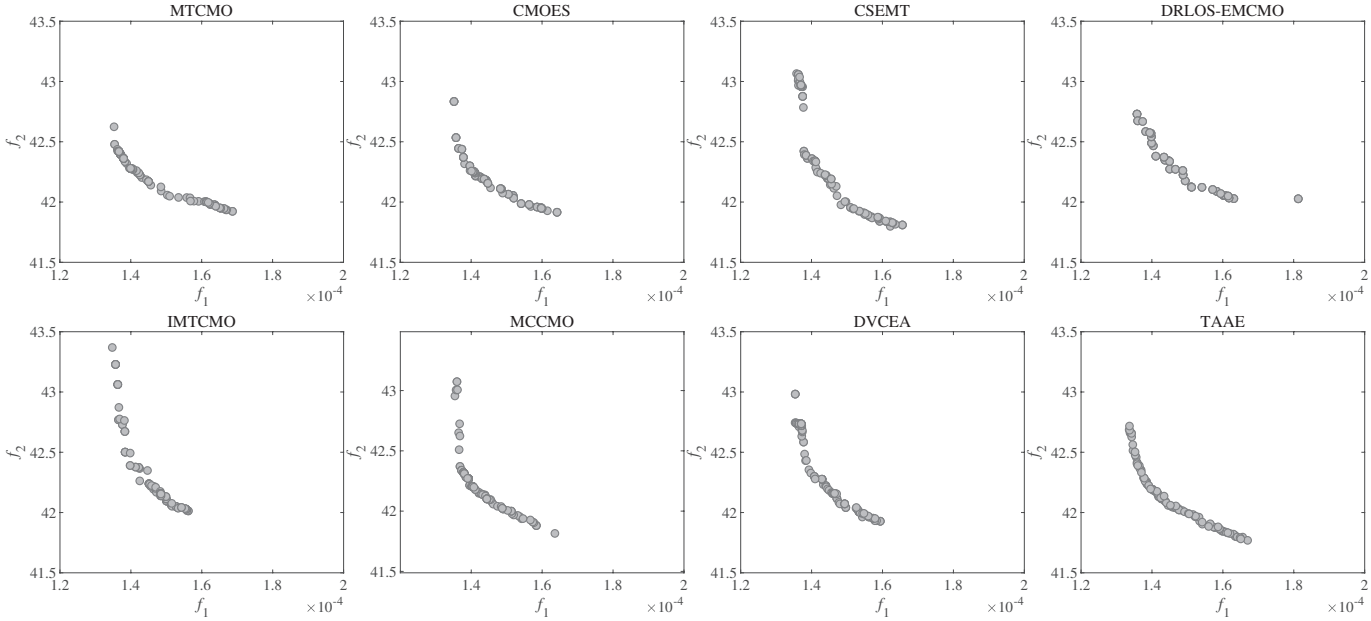


Fig. S3: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS5.

TABLE S3: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C2 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	4.2068e-2 (1.87e-2) $-$	7.2149e-2 (2.67e-2) $-$	3.2135e-2 (2.06e-2) $\approx$	9.7161e-2 (3.33e-2) $-$	8.3541e-2 (3.64e-2) $-$	2.8630e-2 (1.20e-2) $\approx$	6.5471e-2 (3.96e-2) $-$	2.2695e-2 (1.04e-2)
WS2	7.6209e-2 (3.62e-2) $-$	1.0264e-1 (3.79e-2) $-$	8.4519e-2 (2.90e-2) $-$	1.8385e-1 (4.56e-2) $-$	1.6056e-1 (3.91e-2) $-$	7.1652e-2 (4.04e-2) $-$	1.1149e-1 (3.90e-2) $-$	2.2631e-2 (1.62e-2)
WS3	9.4722e-2 (5.31e-2) $-$	1.2944e-1 (6.25e-2) $-$	9.6745e-2 (5.04e-2) $-$	1.8434e-1 (4.79e-2) $-$	1.5551e-1 (3.75e-2) $-$	1.0640e-1 (5.55e-2) $-$	1.4589e-1 (4.27e-2) $-$	4.2762e-2 (4.74e-2)
WS4	1.1284e-1 (2.93e-2) $-$	1.3010e-1 (3.96e-2) $-$	1.0021e-1 (3.52e-2) $\approx$	1.6944e-1 (4.67e-2) $-$	1.4097e-1 (3.06e-2) $-$	1.0109e-1 (2.40e-2) $-$	1.1026e-1 (3.28e-2) $-$	8.4316e-2 (2.75e-2)
WS5	1.2908e-1 (4.56e-2) $-$	1.4566e-1 (4.10e-2) $-$	1.2096e-1 (3.88e-2) $-$	1.9670e-1 (3.82e-2) $-$	1.8552e-1 (3.35e-2) $-$	1.3550e-1 (3.94e-2) $-$	1.6068e-1 (4.17e-2) $-$	9.1097e-2 (2.72e-2)
WS6	5.4510e-2 (3.53e-2) $-$	9.6804e-2 (4.12e-2) $-$	7.6001e-2 (4.07e-2) $-$	1.4053e-1 (2.63e-2) $-$	1.1750e-1 (3.49e-2) $-$	5.5315e-2 (3.16e-2) $-$	8.8704e-2 (3.22e-2) $-$	1.3220e-2 (2.29e-2)
WS7	4.8845e-2 (4.79e-2) $\approx$	1.0022e-1 (4.72e-2) $-$	7.4629e-2 (5.39e-2) $-$	1.1757e-1 (3.52e-2) $-$	8.2635e-2 (4.18e-2) $-$	5.3065e-2 (4.29e-2) $\approx$	7.1386e-2 (3.36e-2) $\approx$	4.9521e-2 (5.50e-2)
WS8	5.8050e-2 (2.91e-2) $-$	1.0149e-1 (3.06e-2) $-$	7.2549e-2 (3.11e-2) $-$	1.4253e-1 (2.73e-2) $-$	1.1269e-1 (3.25e-2) $-$	6.3700e-2 (2.92e-2) $-$	9.2440e-2 (2.71e-2) $-$	1.7532e-2 (1.52e-2)
$+/-/\approx$	0/7/1	0/8/0	0/6/2	0/8/0	0/8/0	0/6/2	0/7/1	

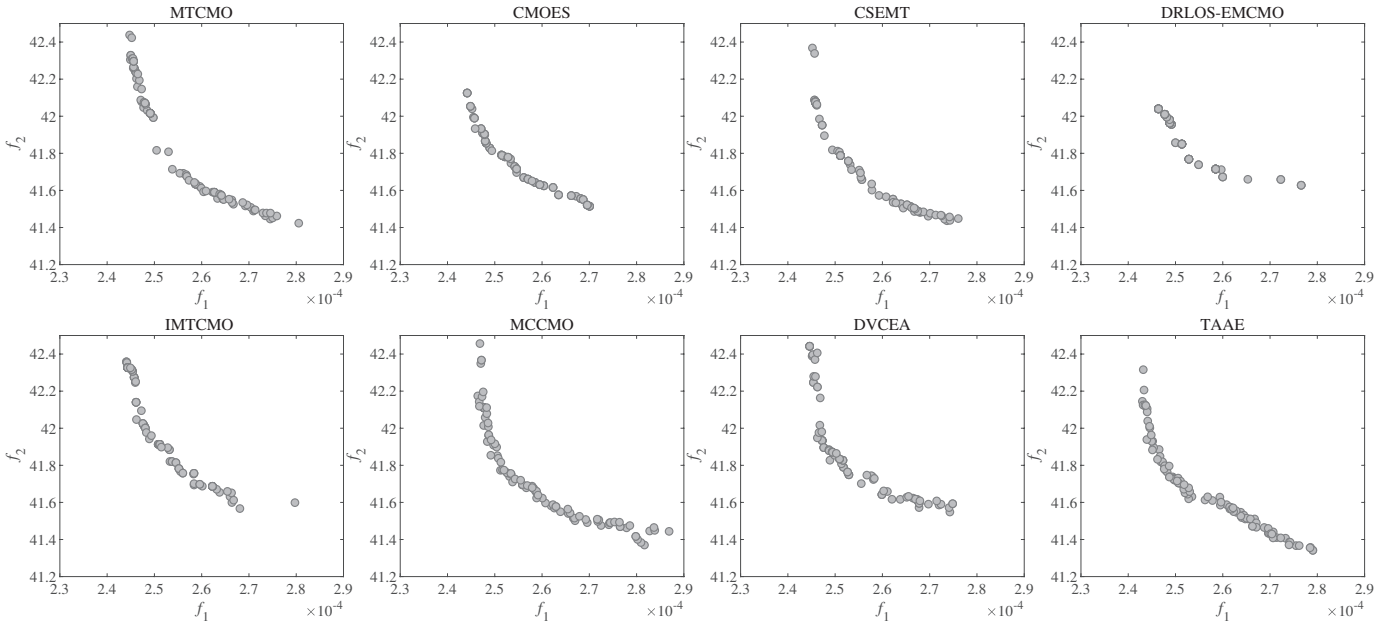


Fig. S4: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS6.

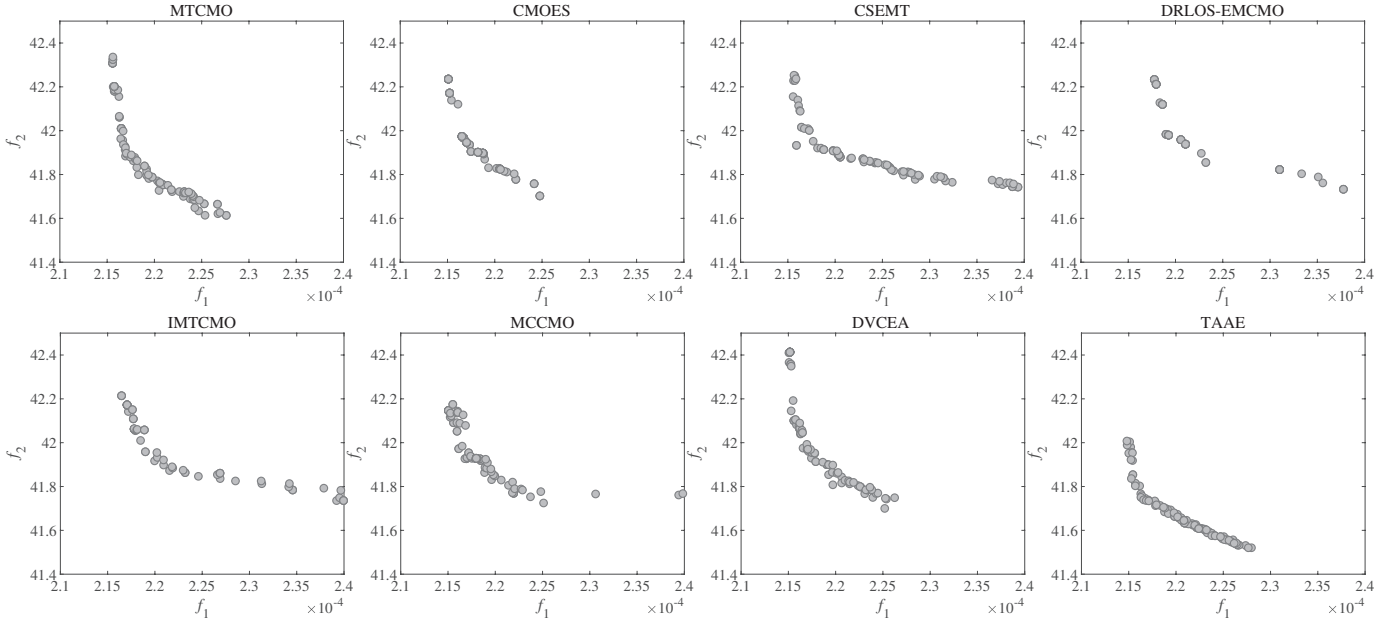


Fig. S5: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS7.

TABLE S4: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C2 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	3.1656e-2 (2.11e-2) $-$	6.4215e-2 (2.72e-2) $-$	2.1455e-2 (2.04e-2) $\approx$	8.7627e-2 (3.40e-2) $-$	7.3995e-2 (3.76e-2) $-$	1.6979e-2 (1.35e-2) $\approx$	5.6520e-2 (4.05e-2) $-$	1.2902e-2 (8.29e-3)
WS2	6.4991e-2 (3.71e-2) $-$	9.5885e-2 (3.85e-2) $-$	7.5400e-2 (2.99e-2) $-$	1.7512e-1 (4.75e-2) $-$	1.5415e-1 (3.96e-2) $-$	6.3465e-2 (4.28e-2) $-$	1.0503e-1 (4.00e-2) $-$	1.4072e-2 (1.63e-2)
WS3	9.0963e-2 (5.38e-2) $-$	1.2561e-1 (6.31e-2) $-$	9.1590e-2 (5.16e-2) $-$	1.8133e-1 (4.87e-2) $-$	1.5168e-1 (3.78e-2) $-$	1.0267e-1 (5.65e-2) $-$	1.4249e-1 (4.35e-2) $-$	3.8305e-2 (4.76e-2)
WS4	5.0871e-2 (3.59e-2) $-$	7.3738e-2 (3.34e-2) $-$	4.7671e-2 (3.64e-2) $-$	1.0987e-1 (4.10e-2) $-$	9.8322e-2 (2.79e-2) $-$	4.4186e-2 (3.59e-2) $-$	5.8496e-2 (2.50e-2) $-$	1.4434e-2 (2.83e-2)
WS5	5.7772e-2 (4.35e-2) $-$	7.4222e-2 (4.33e-2) $-$	5.1683e-2 (3.61e-2) $-$	1.2251e-1 (3.99e-2) $-$	1.2355e-1 (3.59e-2) $-$	6.4346e-2 (4.06e-2) $-$	9.7794e-2 (3.49e-2) $-$	1.4513e-2 (2.85e-2)
WS6	4.9502e-2 (3.60e-2) $-$	9.2907e-2 (4.24e-2) $-$	7.1059e-2 (4.21e-2) $-$	1.3700e-1 (2.74e-2) $-$	1.1357e-1 (3.58e-2) $-$	5.0420e-2 (3.27e-2) $-$	8.4289e-2 (3.32e-2) $-$	9.1046e-3 (2.36e-2)
WS7	4.2071e-2 (4.99e-2) $\approx$	9.5759e-2 (4.88e-2) $-$	6.8745e-2 (5.63e-2) $-$	1.1321e-1 (3.65e-2) $-$	7.7971e-2 (4.34e-2) $-$	4.6208e-2 (4.58e-2) $\approx$	6.4392e-2 (3.46e-2) $\approx$	4.4830e-2 (5.68e-2)
WS8	5.0309e-2 (3.07e-2) $-$	9.2619e-2 (2.85e-2) $-$	6.5078e-2 (3.38e-2) $-$	1.3543e-1 (2.79e-2) $-$	1.0716e-1 (3.36e-2) $-$	5.3853e-2 (3.17e-2) $-$	8.7491e-2 (2.79e-2) $-$	9.5836e-3 (1.62e-2)
$+/-/\approx$	0/7/1	0/8/0	0/7/1	0/8/0	0/8/0	0/6/2	0/7/1	

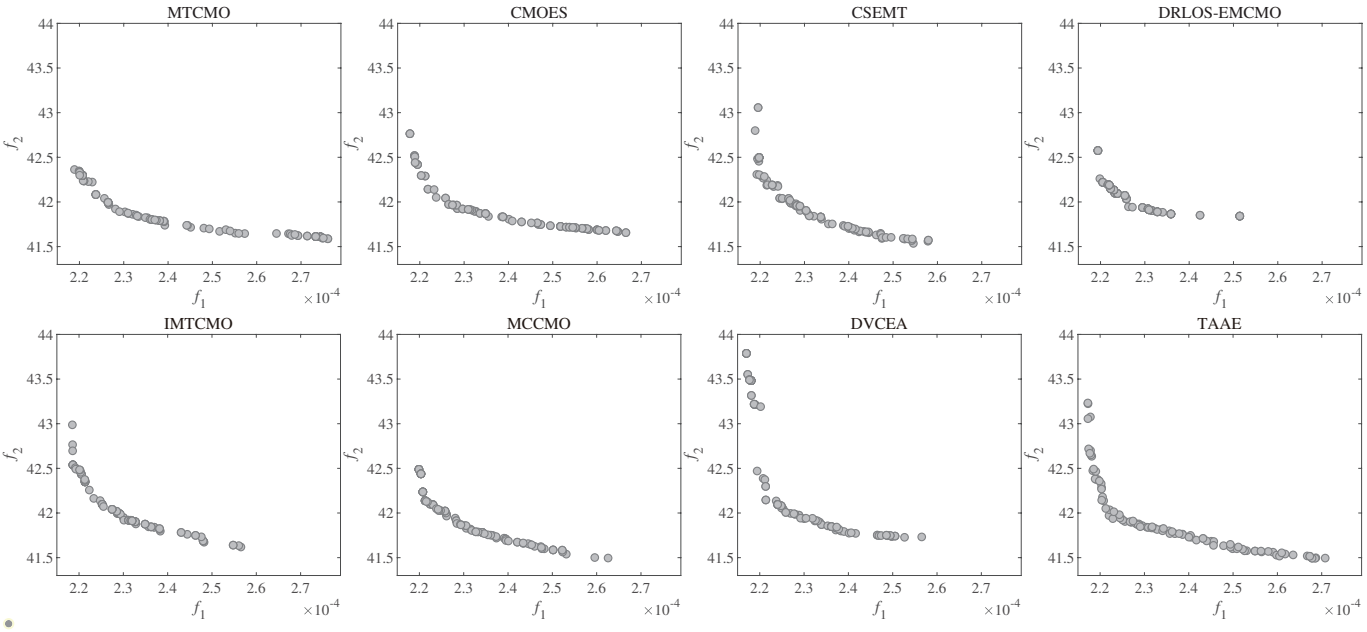


Fig. S6: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C1 test suite under WS8.

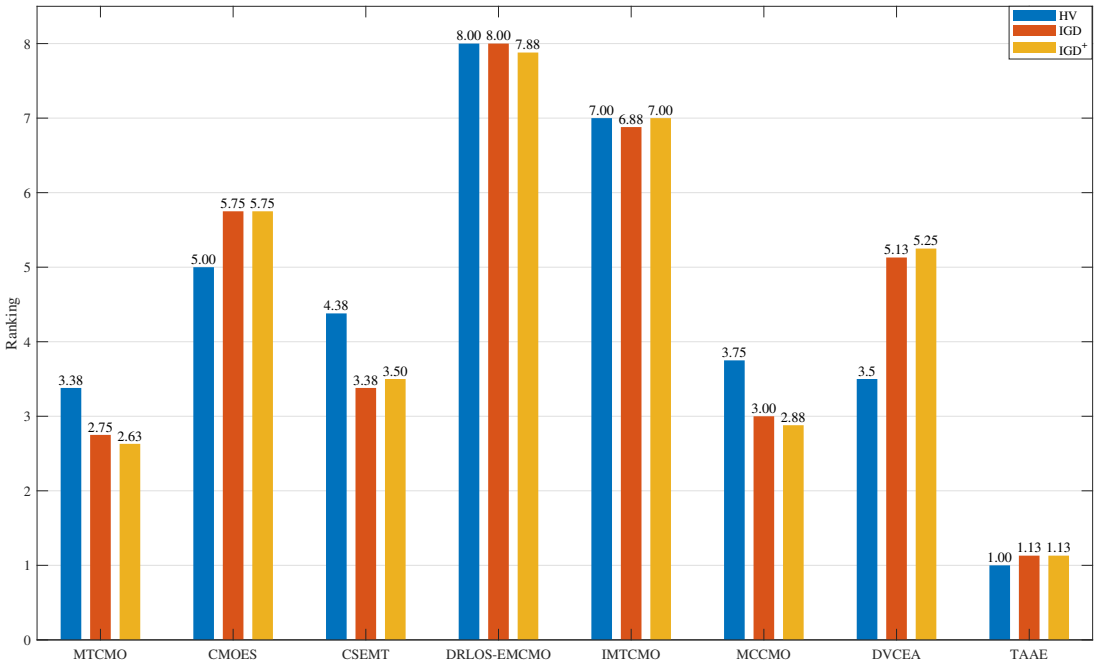


Fig. S7: Friedman test results for TAAE and other compared algorithms on the C2 test suite.

TABLE S5: STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C3 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	8.7910e-2 (1.50e-3) —	8.6043e-2 (1.38e-3) —	8.6997e-2 (1.57e-3) —	8.5165e-2 (1.14e-3) —	8.5090e-2 (1.01e-3) —	8.7374e-2 (1.48e-3) —	8.7029e-2 (1.35e-3) —	8.9560e-2 (1.29e-3)
WS2	7.3813e-2 (1.42e-3) —	7.3216e-2 (1.19e-3) —	7.3456e-2 (1.40e-3) —	7.0920e-2 (1.20e-3) —	7.2202e-2 (1.05e-3) —	7.3840e-2 (1.27e-3) —	7.3514e-2 (1.27e-3) —	7.5699e-2 (1.43e-3)
WS3	3.1252e-2 (8.00e-4) —	3.0828e-2 (4.01e-4) —	3.1156e-2 (7.65e-4) —	2.9866e-2 (6.33e-4) —	3.0697e-2 (4.67e-4) —	3.1109e-2 (7.29e-4) —	3.1041e-2 (3.68e-4) —	3.1985e-2 (8.69e-4)
WS4	3.2772e-2 (5.15e-4) —	3.2712e-2 (6.27e-4) —	3.2607e-2 (6.06e-4) —	3.1574e-2 (5.58e-4) —	3.2143e-2 (5.68e-4) —	3.2417e-2 (6.20e-4) —	3.2759e-2 (4.67e-4) —	3.3517e-2 (6.37e-4)
WS5	3.0995e-2 (6.26e-4) ≈	3.0478e-2 (8.50e-4) —	3.0646e-2 (6.61e-4) —	2.9250e-2 (6.62e-4) —	2.9878e-2 (4.48e-4) —	3.0674e-2 (8.32e-4) —	3.0512e-2 (5.54e-4) —	3.1355e-2 (9.43e-4)
WS6	2.4519e-2 (4.38e-4) —	2.4408e-2 (5.44e-4) —	2.4864e-2 (2.17e-4) ≈	2.3534e-2 (5.34e-4) —	2.3930e-2 (4.34e-4) —	2.4580e-2 (7.59e-4) —	2.4566e-2 (3.21e-4) —	2.5171e-2 (7.49e-4)
WS7	1.5662e-2 (3.21e-4) —	1.5525e-2 (3.81e-4) —	1.5438e-2 (4.03e-4) —	1.5183e-2 (2.89e-4) —	1.5076e-2 (2.27e-4) —	1.5554e-2 (4.08e-4) —	1.5724e-2 (2.15e-4) ≈	1.5990e-2 (4.78e-4)
WS8	4.1464e-2 (8.39e-4) —	4.1391e-2 (6.79e-4) —	4.1283e-2 (8.12e-4) —	4.0308e-2 (5.83e-4) —	4.1038e-2 (6.62e-4) —	4.1377e-2 (7.92e-4) —	4.1685e-2 (5.88e-4) —	4.2235e-2 (9.90e-4)
+/ − / ≈	0/7/1	0/8/0	0/7/1	0/8/0	0/8/0	0/8/0	0/7/1	

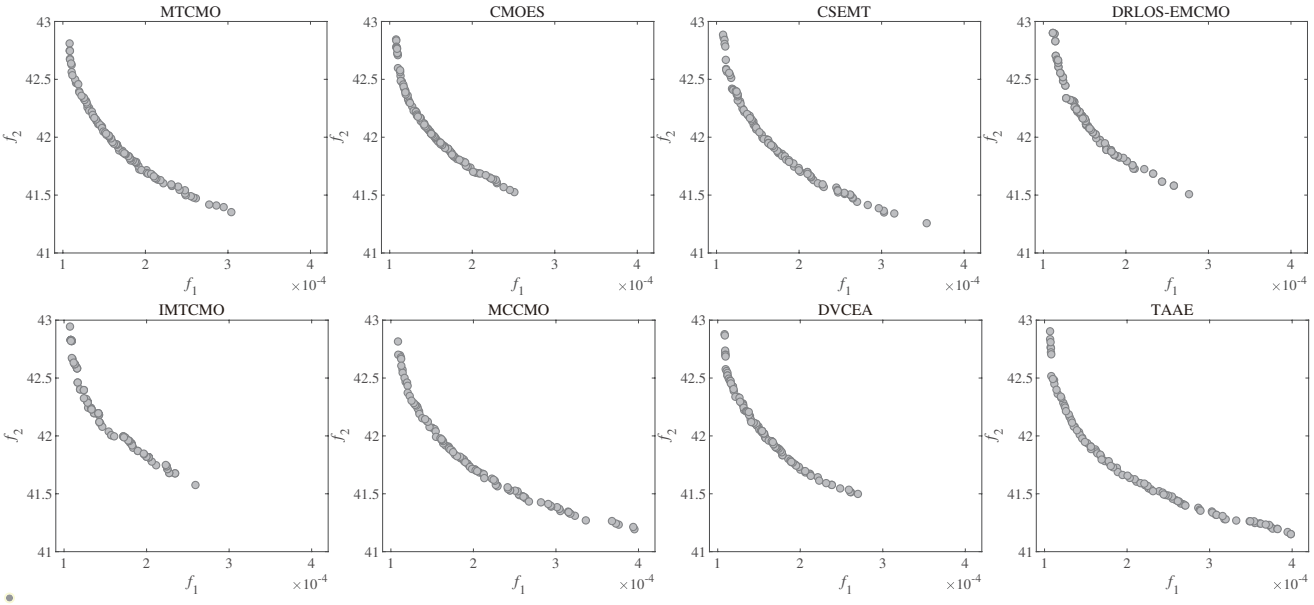


Fig. S8: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS1.

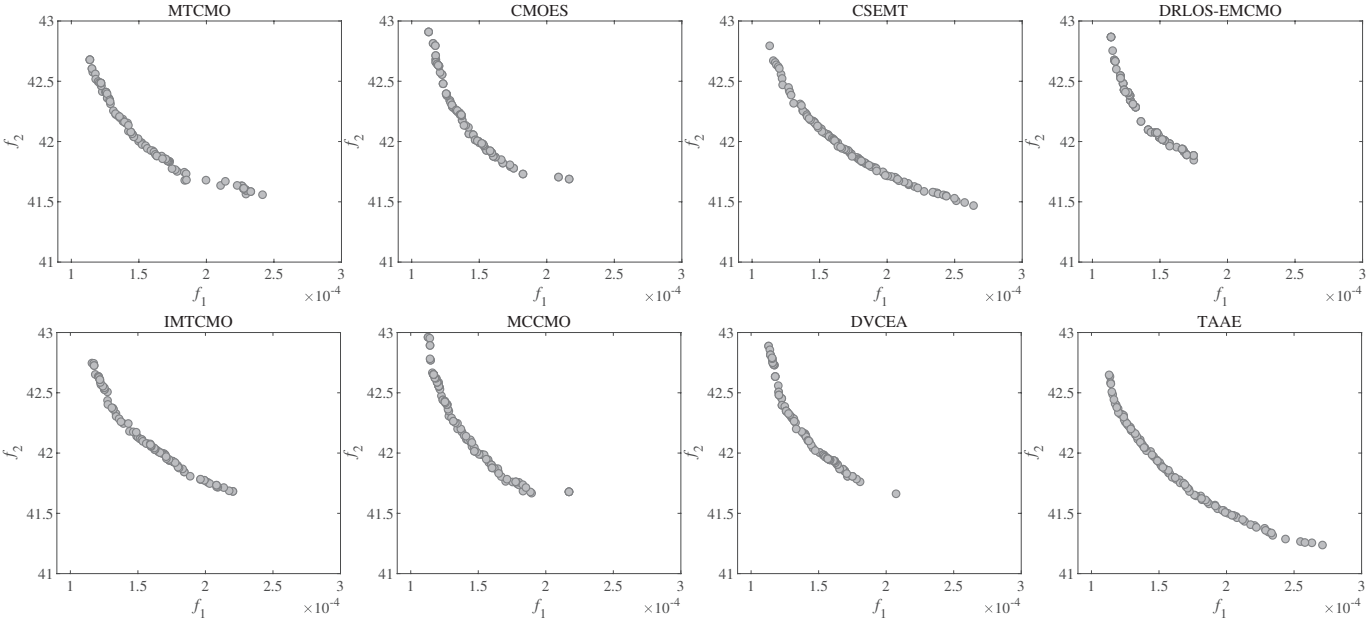


Fig. S9: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS2.

TABLE S6: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C3 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	3.5962e-2 (3.50e-2) ≈	9.4636e-2 (5.07e-2) —	6.7501e-2 (4.40e-2) —	1.0408e-1 (4.21e-2) —	1.0987e-1 (4.78e-2) —	5.2474e-2 (3.96e-2) —	6.6881e-2 (3.31e-2) —	1.7167e-2 (1.01e-2)
WS2	7.3245e-2 (3.66e-2) —	9.3898e-2 (2.94e-2) —	7.7784e-2 (3.68e-2) —	1.6965e-1 (5.53e-2) —	1.5474e-1 (4.22e-2) —	7.2704e-2 (4.20e-2) —	1.0447e-1 (4.43e-2) —	2.6584e-2 (1.76e-2)
WS3	6.9787e-2 (3.74e-2) —	1.0098e-1 (3.38e-2) —	5.3055e-2 (2.64e-2) ≈	1.2643e-1 (2.49e-2) —	1.1461e-1 (2.54e-2) —	7.2881e-2 (3.43e-2) —	8.8751e-2 (2.87e-2) —	3.9857e-2 (3.10e-2)
WS4	6.4413e-2 (3.51e-2) —	8.6687e-2 (3.33e-2) —	7.3296e-2 (3.50e-2) —	1.2086e-1 (2.96e-2) —	1.1565e-1 (3.14e-2) —	7.8782e-2 (2.61e-2) —	7.3341e-2 (2.57e-2) —	4.4026e-2 (2.77e-2)
WS5	4.6438e-2 (3.03e-2) —	7.4743e-2 (4.00e-2) —	4.8428e-2 (3.40e-2) —	1.2896e-1 (4.45e-2) —	9.9751e-2 (2.59e-2) —	5.7952e-2 (4.17e-2) —	8.1115e-2 (3.80e-2) —	2.8261e-2 (2.93e-2)
WS6	5.6850e-2 (4.39e-2) —	8.5440e-2 (3.55e-2) —	3.2510e-2 (2.54e-2) ≈	1.2351e-1 (4.83e-2) —	1.1114e-1 (4.05e-2) —	5.6197e-2 (3.81e-2) —	7.8844e-2 (3.38e-2) —	3.5667e-2 (4.12e-2)
WS7	4.9884e-2 (4.35e-2) ≈	8.3774e-2 (5.27e-2) —	7.6418e-2 (5.16e-2) —	1.2517e-1 (5.03e-2) —	1.1836e-1 (5.21e-2) —	6.8470e-2 (5.90e-2) —	6.9343e-2 (3.63e-2) ≈	4.9176e-2 (5.90e-2)
WS8	1.2451e-1 (2.89e-2) —	1.5044e-1 (2.00e-2) —	1.1690e-1 (2.95e-2) —	1.7429e-1 (3.36e-2) —	1.5241e-1 (2.40e-2) —	1.1280e-1 (2.82e-2) —	1.2980e-1 (2.76e-2) —	9.7390e-2 (2.88e-2)
$+/-/\approx$	0/6/2	0/8/0	0/6/2	0/8/0	0/8/0	0/8/0	0/7/1	

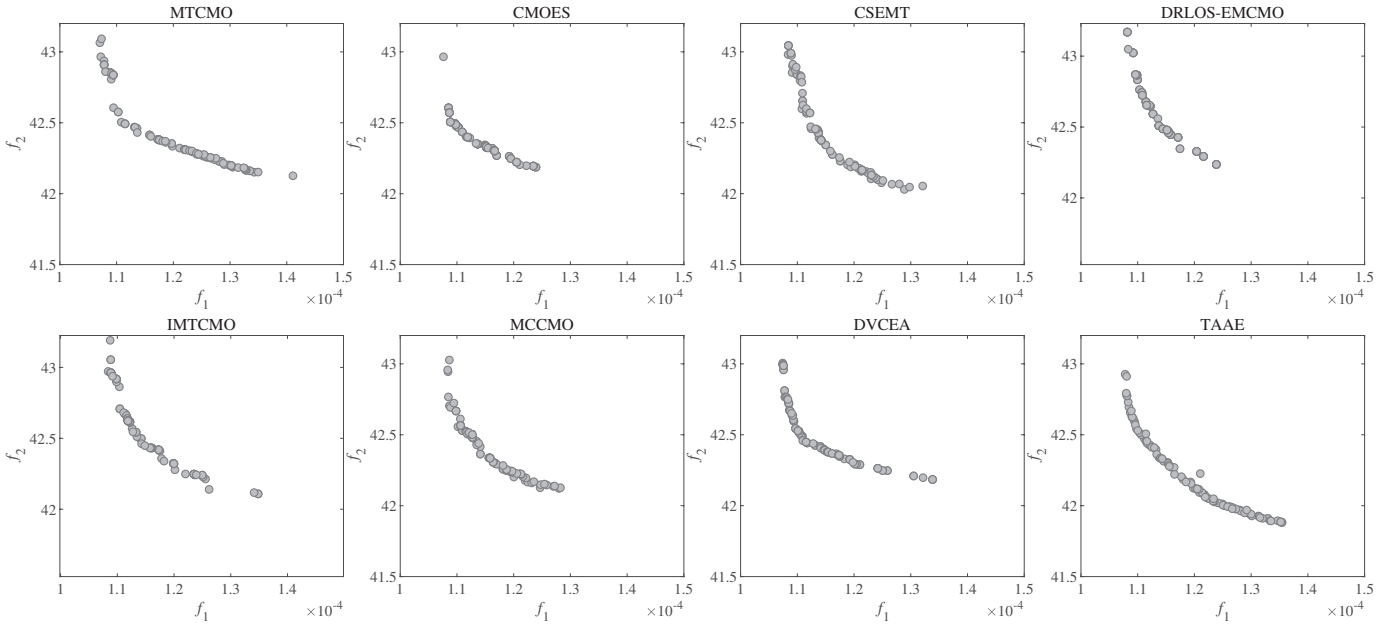


Fig. S10: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS3.

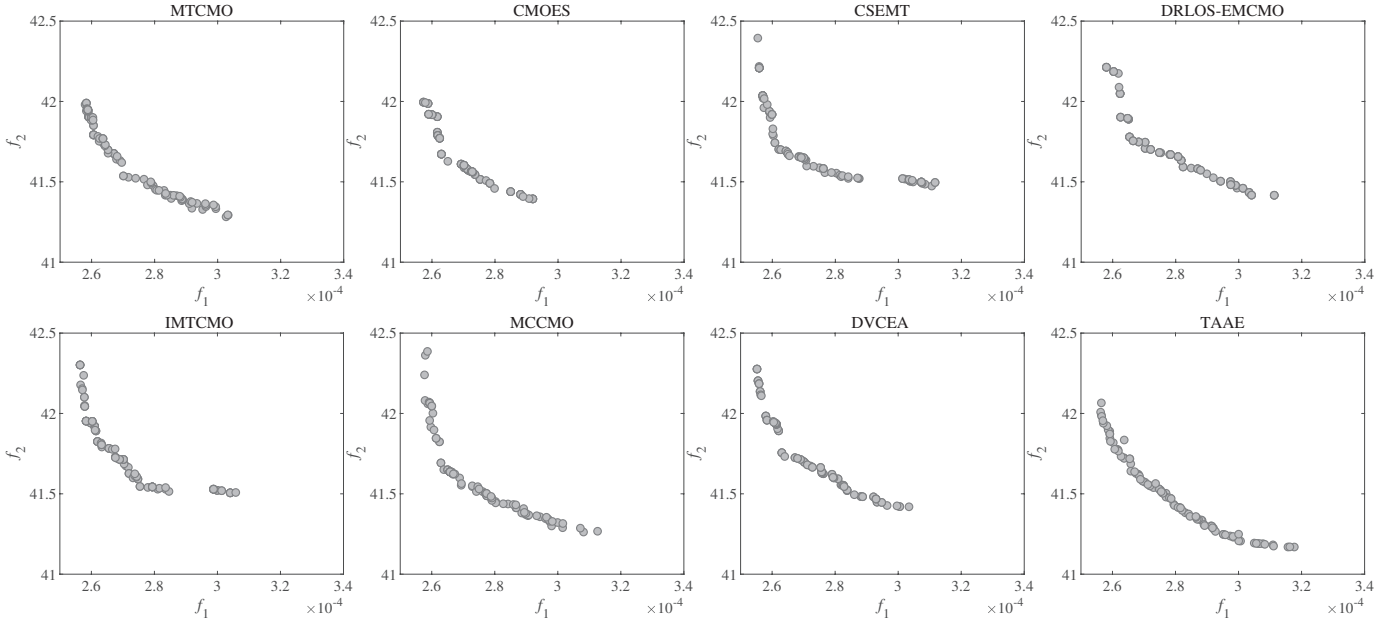


Fig. S11: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS4.

TABLE S7: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C3 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	2.9446e-2 (3.62e-2) ≈	8.9802e-2 (5.17e-2) −	6.1620e-2 (4.54e-2) −	9.6955e-2 (4.34e-2) −	1.0445e-1 (4.91e-2) −	4.7000e-2 (4.02e-2) −	5.9690e-2 (3.40e-2) −	1.1149e-2 (9.87e-3)
WS2	6.7644e-2 (3.78e-2) −	8.8264e-2 (3.04e-2) −	7.2375e-2 (3.75e-2) −	1.6476e-1 (5.67e-2) −	1.4998e-1 (4.27e-2) −	6.6631e-2 (4.34e-2) −	9.9910e-2 (4.51e-2) −	2.1097e-2 (1.81e-2)
WS3	6.1393e-2 (3.90e-2) −	9.3568e-2 (3.56e-2) −	4.4158e-2 (2.90e-2) −	1.1831e-1 (2.68e-2) −	1.0889e-1 (2.63e-2) −	6.3342e-2 (3.74e-2) −	8.1968e-2 (3.02e-2) −	3.0804e-2 (3.29e-2)
WS4	4.0822e-2 (3.85e-2) −	5.8466e-2 (3.22e-2) −	4.9016e-2 (3.95e-2) −	9.3639e-2 (3.28e-2) −	9.4676e-2 (3.52e-2) −	5.1396e-2 (2.52e-2) −	4.6997e-2 (2.63e-2) −	1.6265e-2 (2.87e-2)
WS5	3.8431e-2 (2.94e-2) −	6.7537e-2 (4.17e-2) −	4.1452e-2 (3.51e-2) −	1.2071e-1 (4.62e-2) −	9.4435e-2 (2.60e-2) −	5.0235e-2 (4.29e-2) −	7.4859e-2 (3.94e-2) −	1.8075e-2 (3.02e-2)
WS6	5.0370e-2 (4.59e-2) ≈	8.0147e-2 (3.54e-2) −	2.5655e-2 (2.61e-2) ≈	1.1905e-1 (4.94e-2) −	1.0666e-1 (4.14e-2) −	5.1684e-2 (3.92e-2) −	7.4533e-2 (3.48e-2) −	3.1233e-2 (4.15e-2)
WS7	4.3232e-2 (4.56e-2) ≈	7.9652e-2 (5.37e-2) −	7.0973e-2 (5.35e-2) −	1.2180e-1 (5.12e-2) −	1.1484e-1 (5.36e-2) −	6.2781e-2 (6.06e-2) −	6.4210e-2 (3.83e-2) ≈	4.4444e-2 (6.08e-2)
WS8	5.1758e-2 (2.66e-2) −	7.9968e-2 (2.18e-2) −	4.2107e-2 (2.82e-2) −	9.9252e-2 (3.80e-2) −	8.7684e-2 (2.52e-2) −	3.7238e-2 (2.68e-2) ≈	6.1069e-2 (2.93e-2) −	2.3555e-2 (2.80e-2)
$+/-/\approx$	0/5/3	0/8/0	0/7/1	0/8/0	0/8/0	0/7/1	0/7/1	

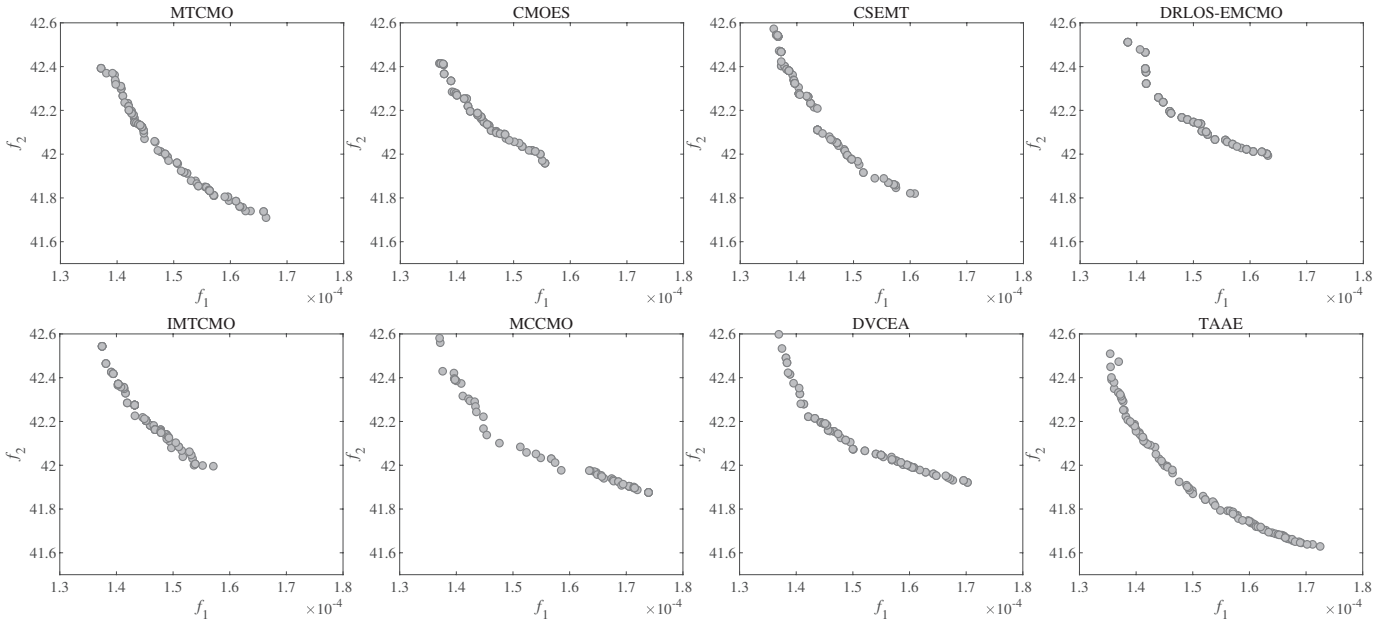


Fig. S12: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS5.

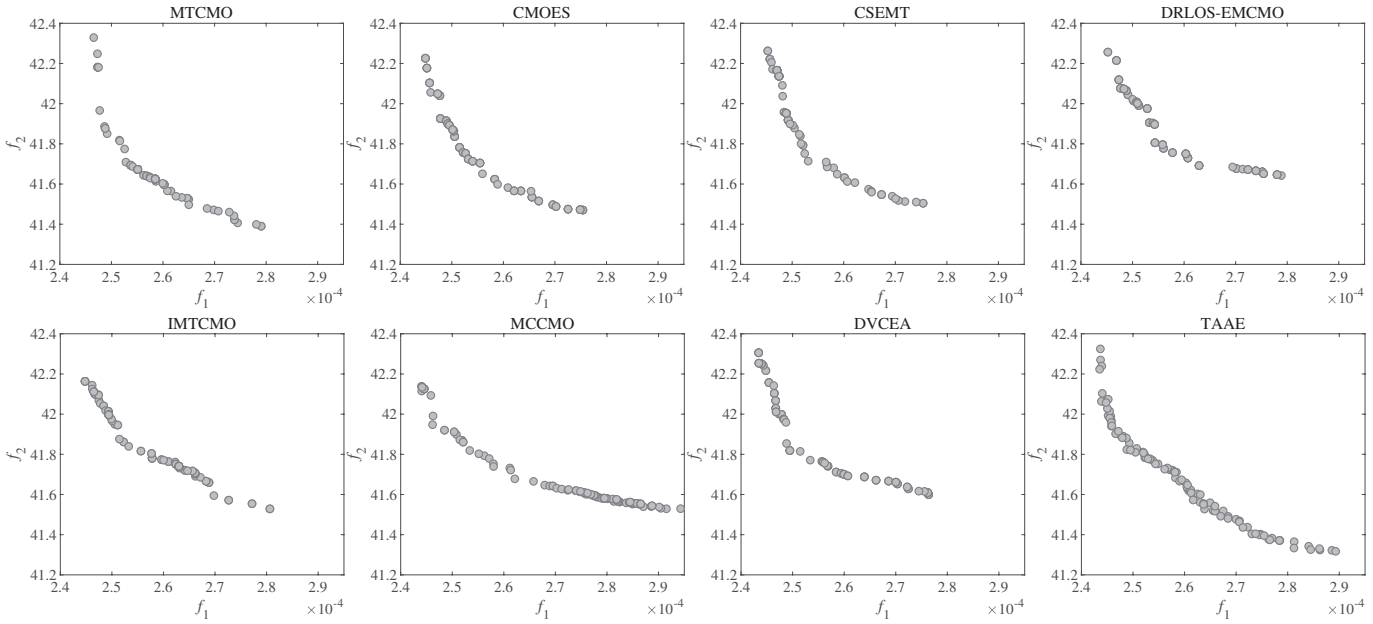


Fig. S13: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS6.

TABLE S8: STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C4 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	7.2533e-2 (2.62e-3) −	7.0583e-2 (2.56e-3) −	7.0767e-2 (3.30e-3) −	6.7619e-2 (5.00e-3) −	6.9389e-2 (2.41e-3) −	7.1752e-2 (2.45e-3) −	7.0326e-2 (2.51e-3) −	7.6565e-2 (1.45e-3)
WS2	1.0272e-1 (2.77e-3) −	1.0195e-1 (3.49e-3) −	1.0075e-1 (4.46e-3) −	9.5354e-2 (4.98e-3) −	1.0123e-1 (2.74e-3) −	1.0083e-1 (4.57e-3) −	1.0096e-1 (3.13e-3) −	1.0694e-1 (1.52e-3)
WS3	3.0565e-2 (1.24e-3) −	2.9759e-2 (1.26e-3) −	2.8831e-2 (1.93e-3) −	2.7723e-2 (2.13e-3) −	2.7612e-2 (1.36e-3) −	2.8996e-2 (2.39e-3) −	2.9786e-2 (1.16e-3) −	3.3006e-2 (4.51e-4)
WS4	4.2177e-2 (1.59e-3) −	4.1577e-2 (1.42e-3) −	4.1312e-2 (2.11e-3) −	4.0095e-2 (2.14e-3) −	4.1029e-2 (1.19e-3) −	4.1551e-2 (1.66e-3) −	4.1685e-2 (1.08e-3) −	4.4497e-2 (4.95e-4)
WS5	4.4640e-2 (1.52e-3) −	4.3221e-2 (1.58e-3) −	4.3088e-2 (1.76e-3) −	4.0331e-2 (2.22e-3) −	4.1816e-2 (2.31e-3) −	4.3752e-2 (2.20e-3) −	4.3442e-2 (1.30e-3) −	4.7022e-2 (6.64e-4)
WS6	4.0146e-2 (1.65e-3) −	3.9577e-2 (1.60e-3) −	4.0351e-2 (1.36e-3) −	3.8670e-2 (2.21e-3) −	3.8516e-2 (1.50e-3) −	4.0428e-2 (1.78e-3) −	3.9602e-2 (1.14e-3) −	4.2680e-2 (5.87e-4)
WS7	2.4614e-2 (1.04e-3) −	2.4321e-2 (1.12e-3) −	2.4478e-2 (9.81e-4) −	2.2934e-2 (1.46e-3) −	2.2888e-2 (1.26e-3) −	2.4257e-2 (1.43e-3) −	2.4118e-2 (1.39e-3) −	2.6625e-2 (3.56e-4)
WS8	3.9534e-2 (1.45e-3) −	3.8034e-2 (1.67e-3) −	3.7672e-2 (2.09e-3) −	3.6861e-2 (2.22e-3) −	3.7481e-2 (1.65e-3) −	3.7702e-2 (2.11e-3) −	3.8357e-2 (1.80e-3) −	4.1195e-2 (5.89e-4)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

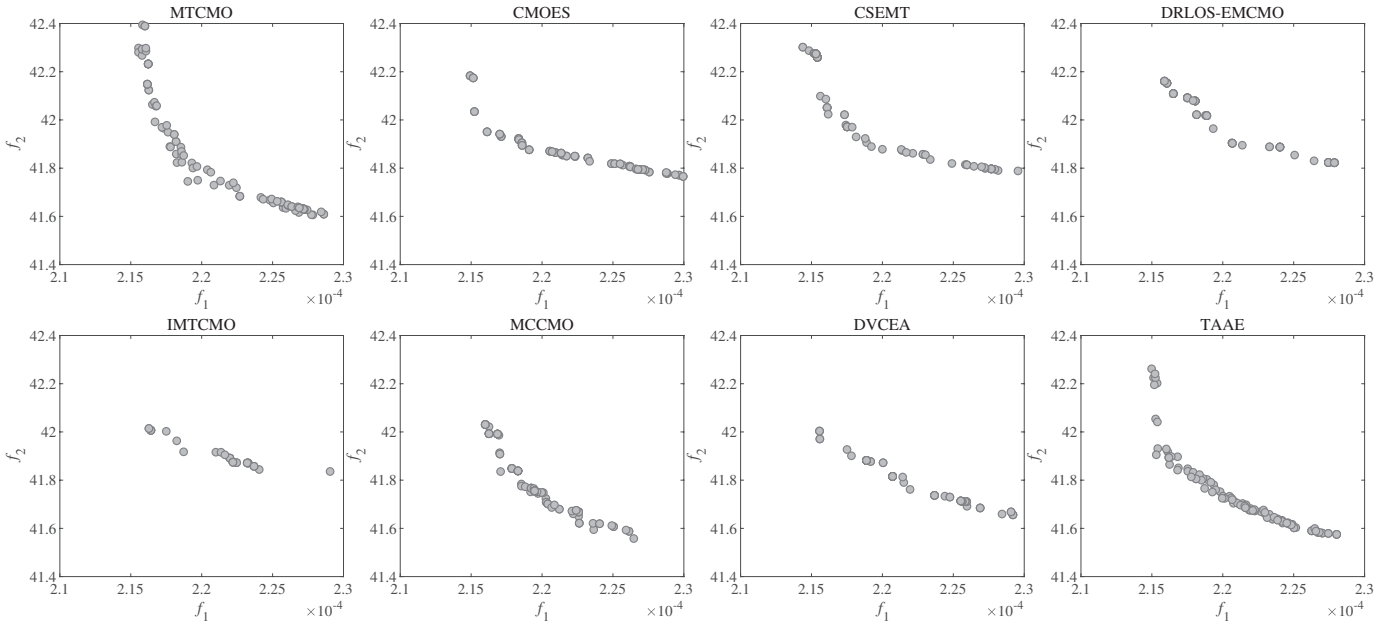


Fig. S14: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS7.

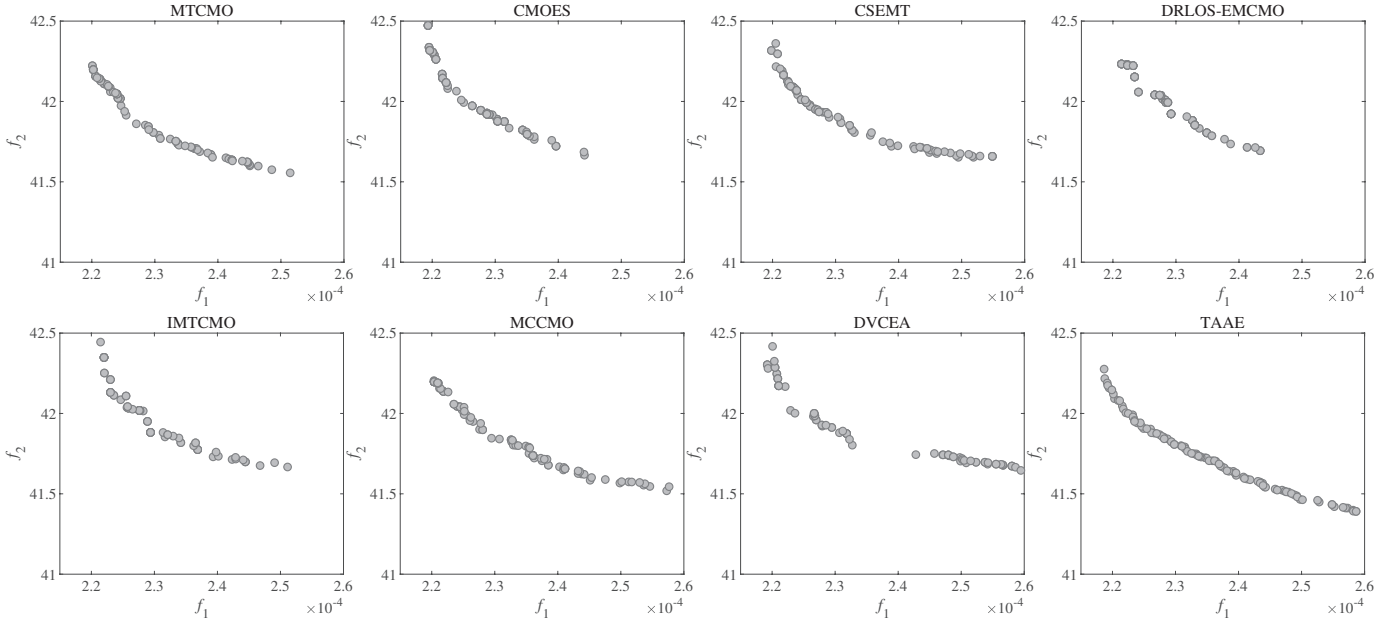


Fig. S15: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C2 test suite under WS8.

TABLE S9: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C4 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	1.4893e-1 (7.41e-2) $-$	2.8595e-1 (1.27e-1) $-$	2.2751e-1 (1.59e-1) $-$	4.2530e-1 (4.02e-1) $-$	2.1799e-1 (1.60e-1) $-$	1.5486e-1 (6.92e-2) $-$	2.3605e-1 (1.40e-1) $-$	8.1677e-2 (3.20e-2)
WS2	1.9884e-1 (9.35e-2) $-$	2.7326e-1 (1.22e-1) $-$	2.9563e-1 (1.83e-1) $-$	5.1666e-1 (3.00e-1) $-$	2.6304e-1 (8.31e-2) $-$	3.0299e-1 (2.13e-1) $-$	2.9771e-1 (1.44e-1) $-$	1.0382e-1 (3.94e-2)
WS3	1.3169e-1 (1.23e-1) $\approx$	2.5681e-1 (1.82e-1) $-$	3.1214e-1 (2.26e-1) $-$	2.5436e-1 (3.31e-1) $\approx$	2.1938e-1 (2.05e-1) $\approx$	2.7375e-1 (2.43e-1) $\approx$	1.4262e-1 (1.44e-1) $\approx$	1.1392e-1 (4.53e-2)
WS4	3.5146e-1 (2.08e-1) $-$	4.4865e-1 (2.04e-1) $-$	3.5851e-1 (2.32e-1) $-$	5.3604e-1 (3.11e-1) $-$	3.3101e-1 (2.02e-1) $-$	3.4196e-1 (1.28e-1) $-$	3.9006e-1 (1.75e-1) $-$	1.7623e-1 (6.29e-2)
WS5	2.0367e-1 (1.18e-1) $-$	4.5006e-1 (2.03e-1) $-$	2.6144e-1 (1.63e-1) $-$	4.8777e-1 (2.86e-1) $-$	3.2871e-1 (2.72e-1) $-$	2.9883e-1 (2.22e-1) $-$	3.2306e-1 (1.90e-1) $-$	9.1513e-2 (2.70e-2)
WS6	2.3914e-1 (1.26e-1) $-$	3.9992e-1 (1.71e-1) $-$	2.2569e-1 (1.15e-1) $-$	3.7671e-1 (3.76e-1) $-$	3.4176e-1 (1.38e-1) $-$	2.6391e-1 (2.12e-1) $-$	3.0272e-1 (1.14e-1) $-$	1.3468e-1 (4.20e-2)
WS7	2.0132e-1 (1.49e-1) $-$	3.1624e-1 (2.09e-1) $-$	2.0243e-1 (1.95e-1) $-$	4.1952e-1 (3.31e-1) $-$	3.2724e-1 (3.00e-1) $-$	2.5598e-1 (2.04e-1) $-$	2.7277e-1 (2.41e-1) $-$	3.9523e-2 (1.17e-2)
WS8	1.4184e-1 (7.14e-2) $-$	2.0512e-1 (1.01e-1) $-$	2.4678e-1 (1.38e-1) $-$	2.7703e-1 (1.94e-1) $-$	1.9623e-1 (1.24e-1) $-$	2.2099e-1 (9.27e-2) $-$	1.8139e-1 (9.93e-2) $-$	7.4239e-2 (1.75e-2)
$+/-/\approx$	0/7/1	0/8/0	0/8/0	0/7/1	0/7/1	0/7/1	0/7/1	

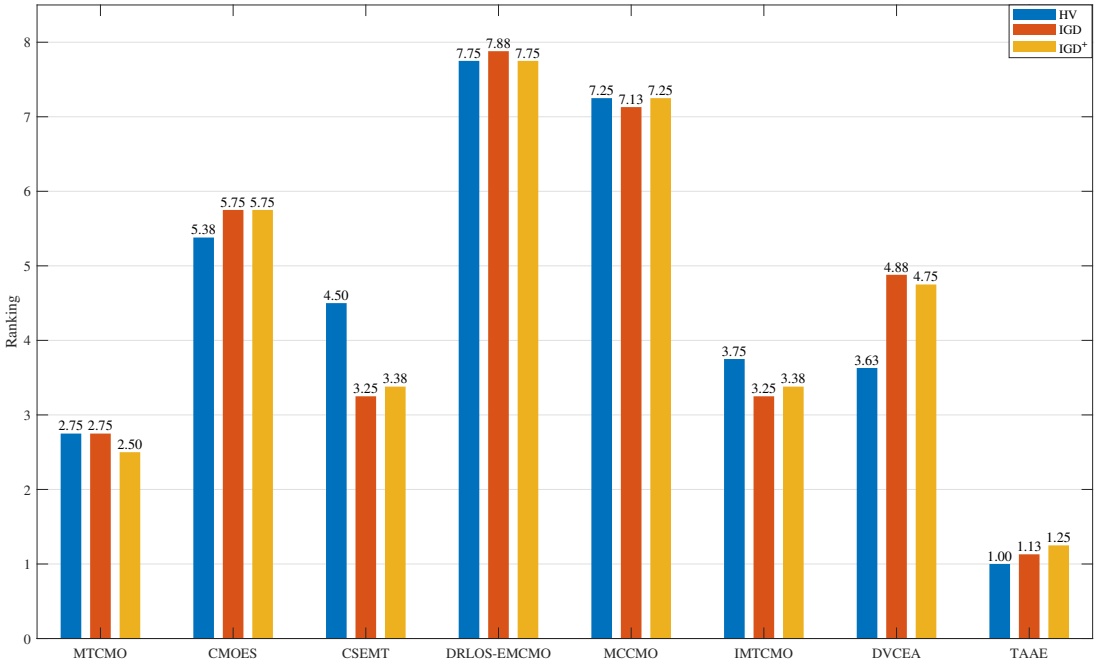


Fig. S16: Friedman test results for TAAE and other compared algorithms on the C3 test suite.

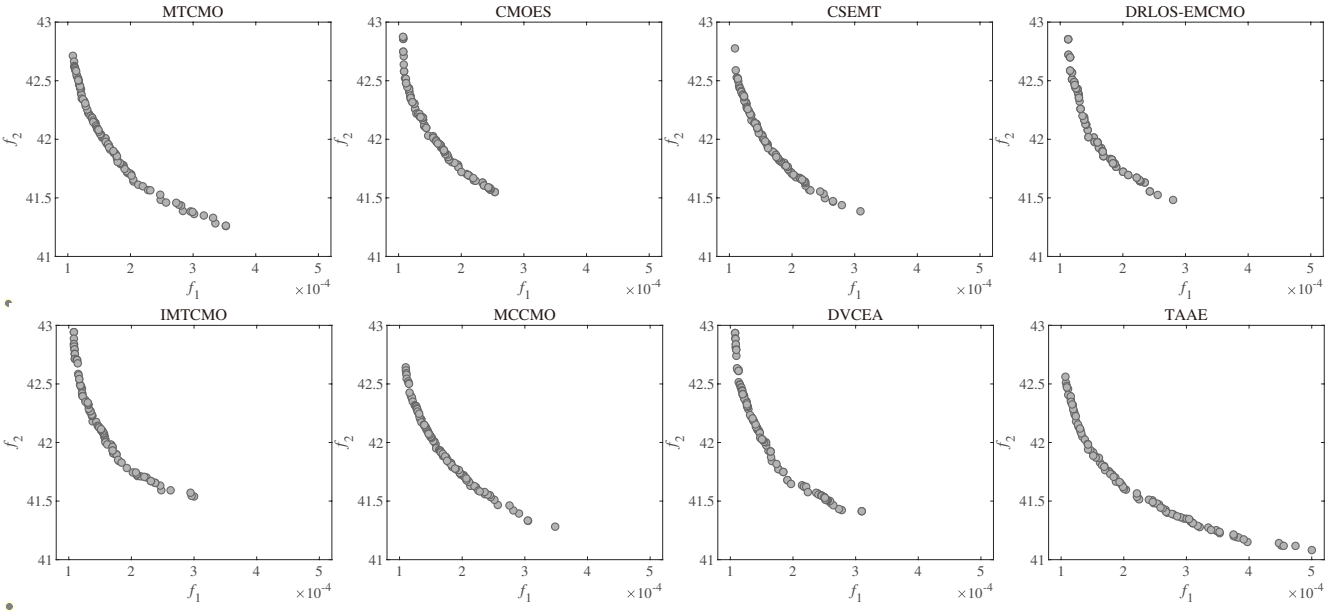


Fig. S17: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS1.

TABLE S10: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C4 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	1.2062e-1 (7.77e-2) —	2.6454e-1 (1.35e-1) —	1.9753e-1 (1.71e-1) —	3.9640e-1 (4.18e-1) —	1.8757e-1 (1.69e-1) —	1.2318e-1 (7.03e-2) —	2.1402e-1 (1.49e-1) —	2.7626e-2 (2.07e-2)
WS2	1.2239e-1 (9.62e-2) —	1.9699e-1 (1.33e-1) —	2.1834e-1 (1.99e-1) —	4.3258e-1 (3.26e-1) —	1.8789e-1 (8.69e-2) —	2.2245e-1 (2.35e-1) —	2.4548e-1 (1.47e-1) —	3.5364e-2 (4.37e-2)
WS3	1.0024e-1 (1.37e-1) —	2.3904e-1 (1.86e-1) —	2.9672e-1 (2.36e-1) —	2.2456e-1 (3.39e-1) —	1.9629e-1 (2.16e-1) —	2.5410e-1 (2.53e-1) —	1.0984e-1 (1.57e-1) —	1.4146e-2 (1.10e-2)
WS4	2.2357e-1 (1.75e-1) —	3.5619e-1 (2.02e-1) —	2.4514e-1 (2.44e-1) —	4.1015e-1 (3.40e-1) —	2.5755e-1 (1.99e-1) —	2.3150e-1 (1.41e-1) —	3.0154e-1 (2.12e-1) —	5.8428e-2 (3.30e-2)
WS5	1.6472e-1 (1.28e-1) ≈	4.2204e-1 (2.11e-1) —	2.1965e-1 (1.70e-1) —	4.5831e-1 (2.97e-1) —	2.9592e-1 (2.82e-1) —	2.6162e-1 (2.33e-1) —	2.9607e-1 (1.96e-1) —	6.9576e-2 (2.97e-2)
WS6	1.8693e-1 (1.44e-1) —	3.5560e-1 (1.82e-1) —	1.6322e-1 (1.22e-1) —	2.9990e-1 (3.97e-1) —	3.0259e-1 (1.50e-1) —	1.9377e-1 (2.25e-1) ≈	2.6338e-1 (1.22e-1) —	8.0390e-2 (4.24e-2)
WS7	1.8550e-1 (1.63e-1) ≈	3.0642e-1 (2.14e-1) —	1.8440e-1 (2.02e-1) —	4.0700e-1 (3.43e-1) —	3.1409e-1 (3.09e-1) —	2.3741e-1 (2.19e-1) —	2.6106e-1 (2.49e-1) —	2.2353e-2 (6.72e-3)
WS8	5.6689e-2 (4.91e-2) —	1.4823e-1 (1.10e-1) —	1.6760e-1 (1.66e-1) —	2.0409e-1 (2.14e-1) —	1.2477e-1 (1.30e-1) —	1.4290e-1 (1.07e-1) —	1.2471e-1 (1.01e-1) —	2.3564e-2 (1.96e-2)
+/ − / ≈	0/6/2	0/8/0	0/8/0	0/8/0	0/8/0	0/7/1	0/8/0	

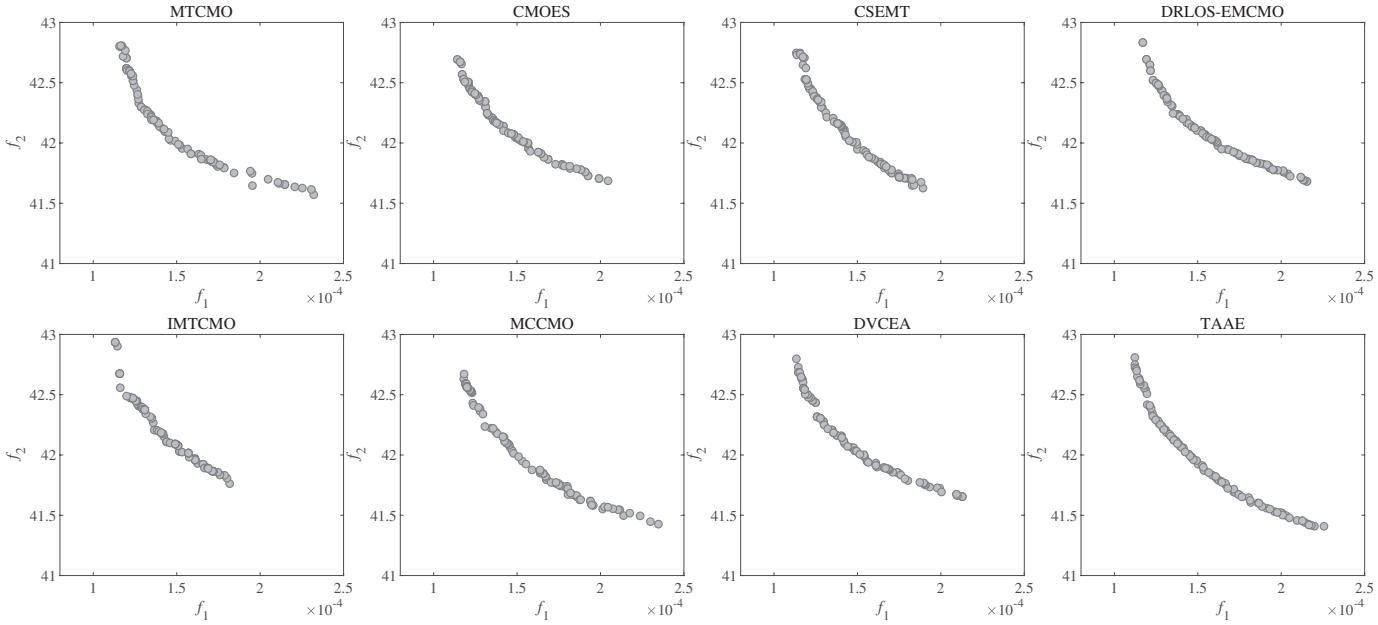


Fig. S18: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS2.

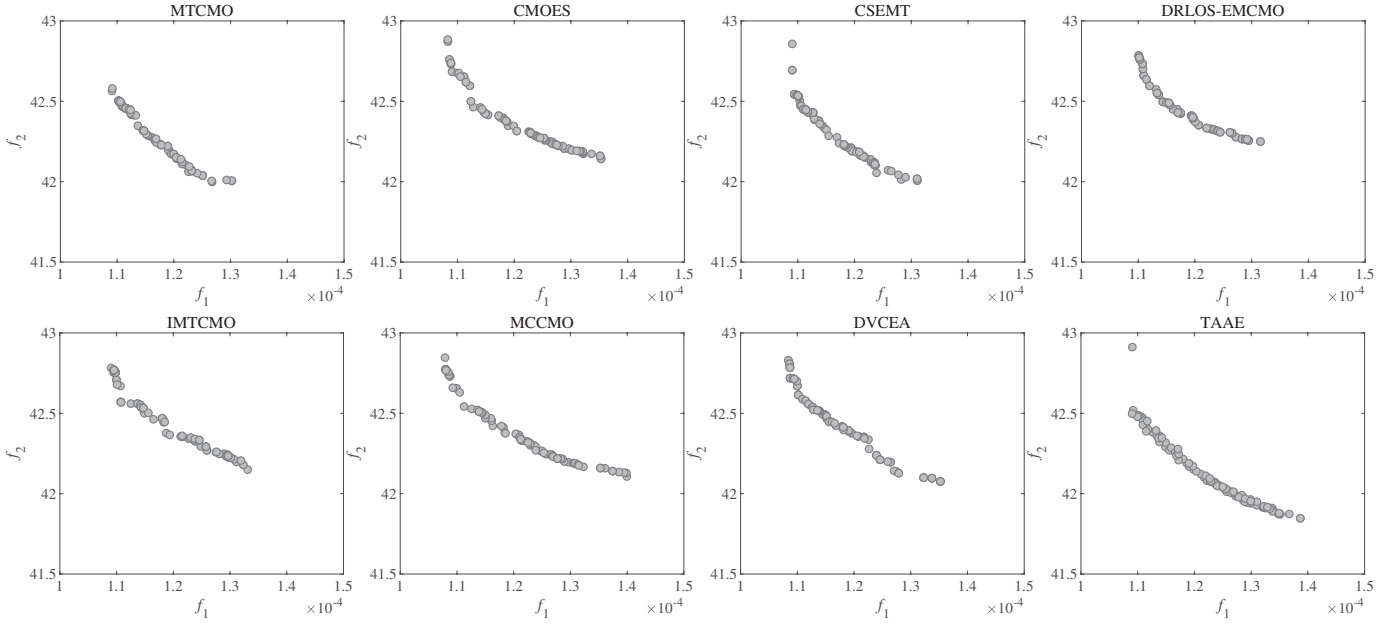


Fig. S19: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS3.

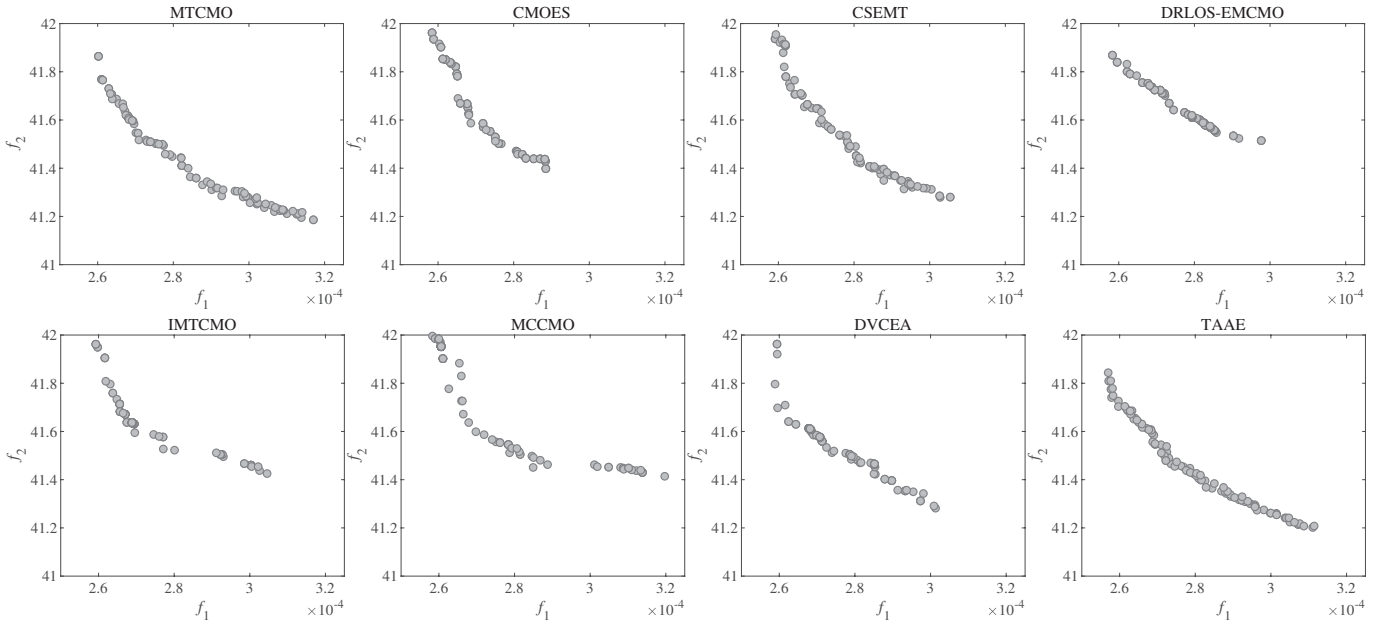


Fig. S20: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS4.

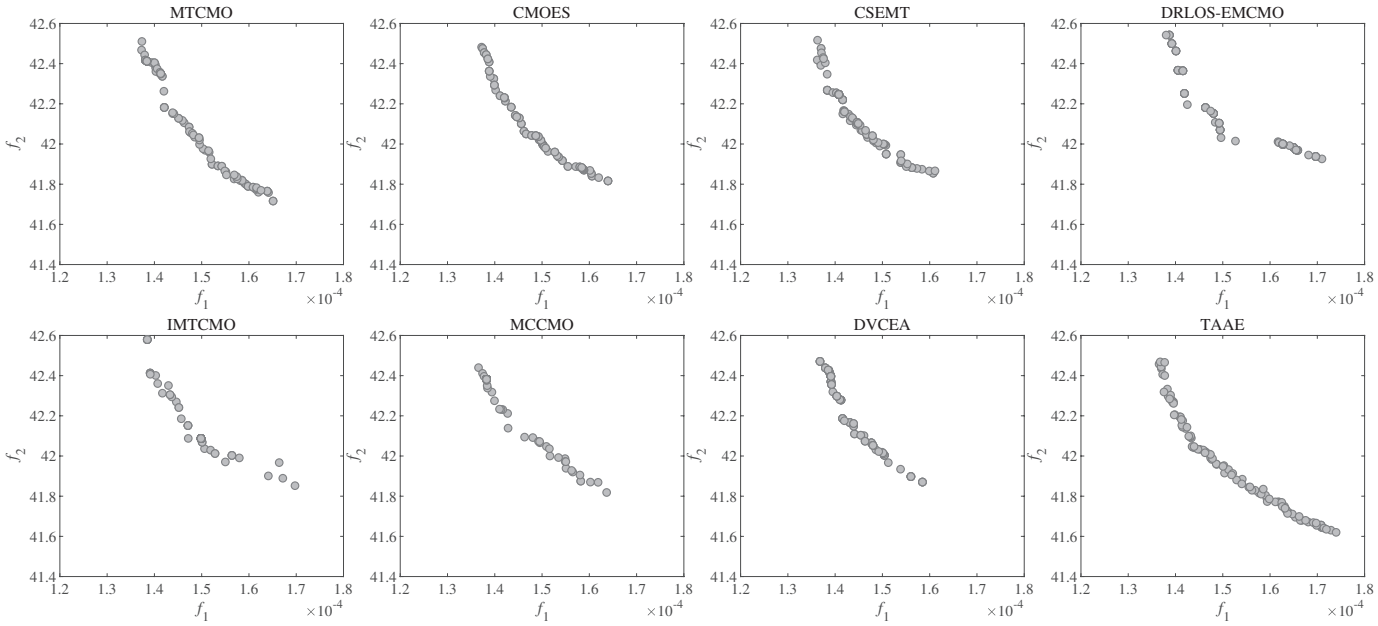


Fig. S21: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS5.

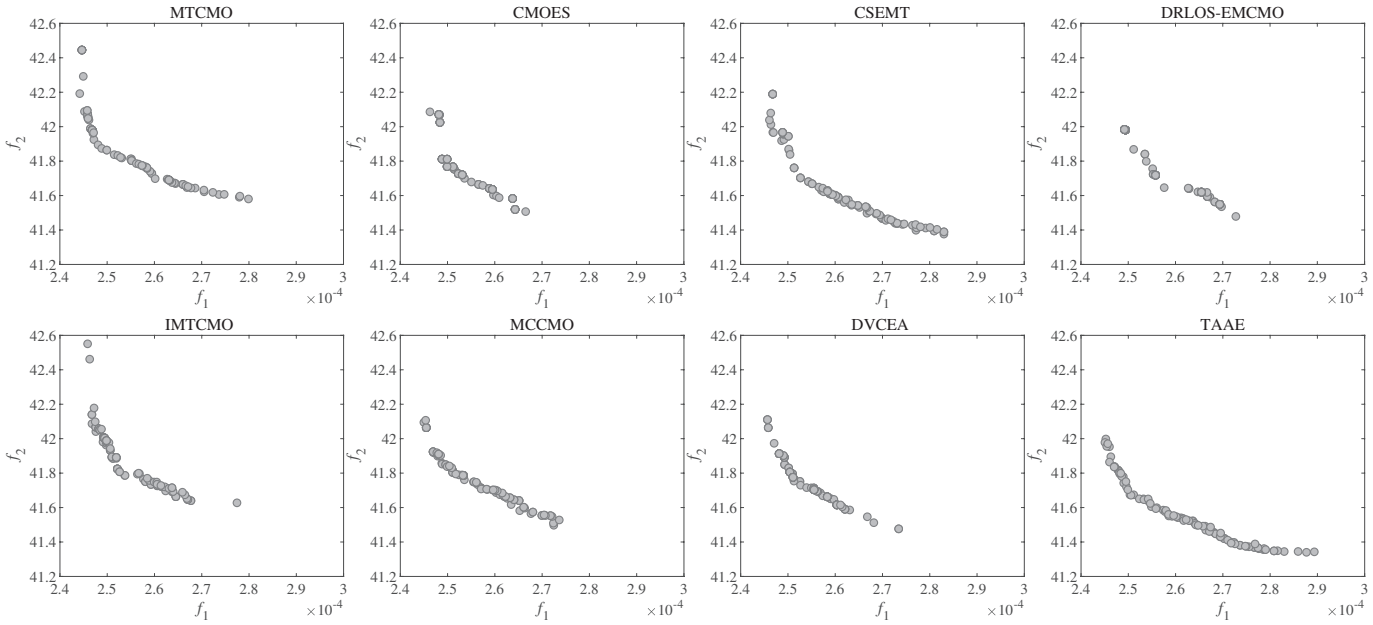


Fig. S22: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS6.

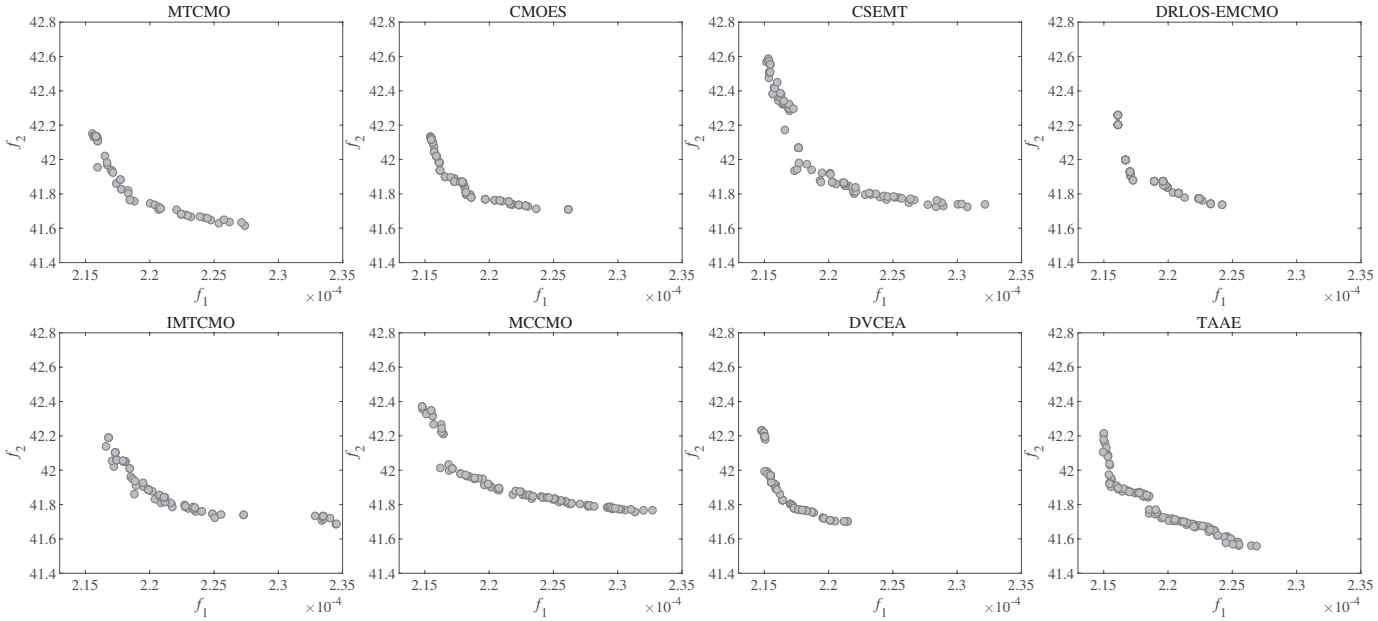


Fig. S23: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS7.

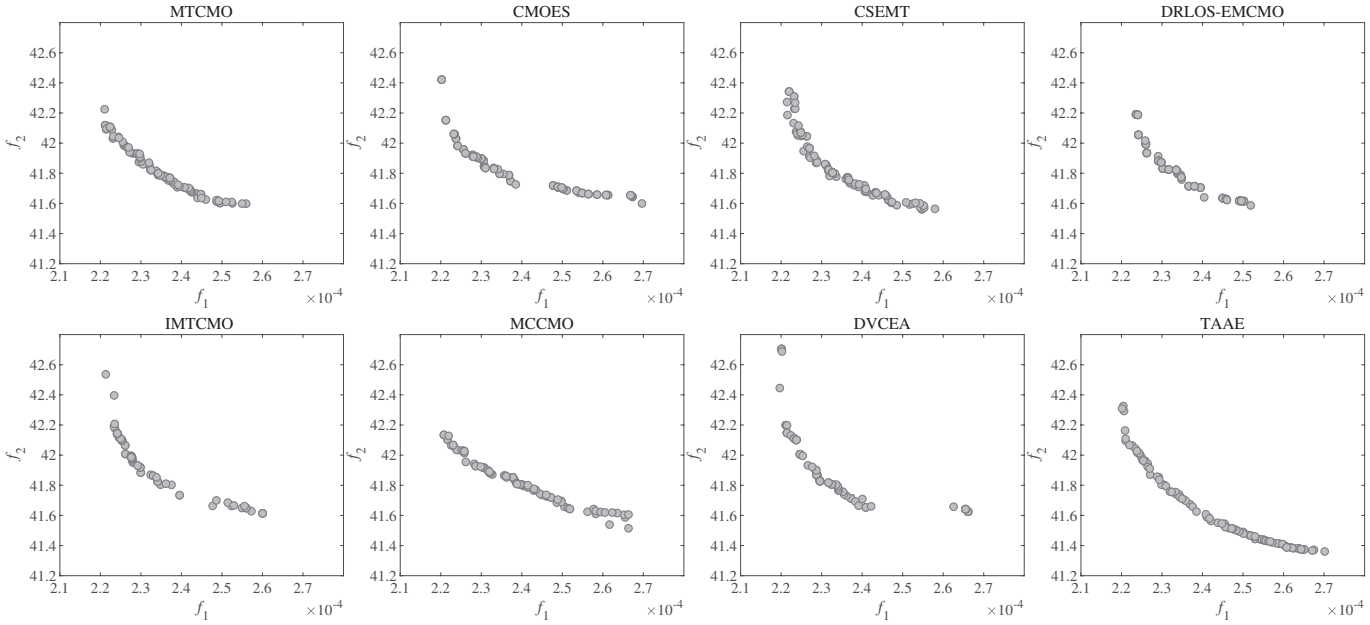


Fig. S24: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C3 test suite under WS8.

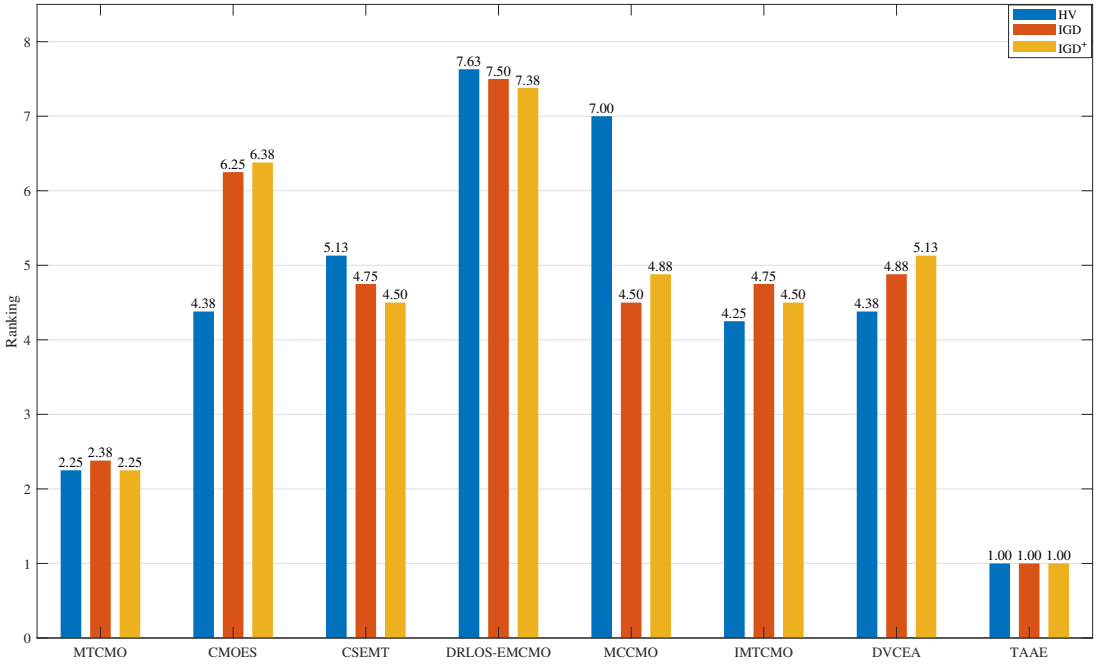


Fig. S25: Friedman test results for TAAE and other compared algorithms on the C4 test suite.

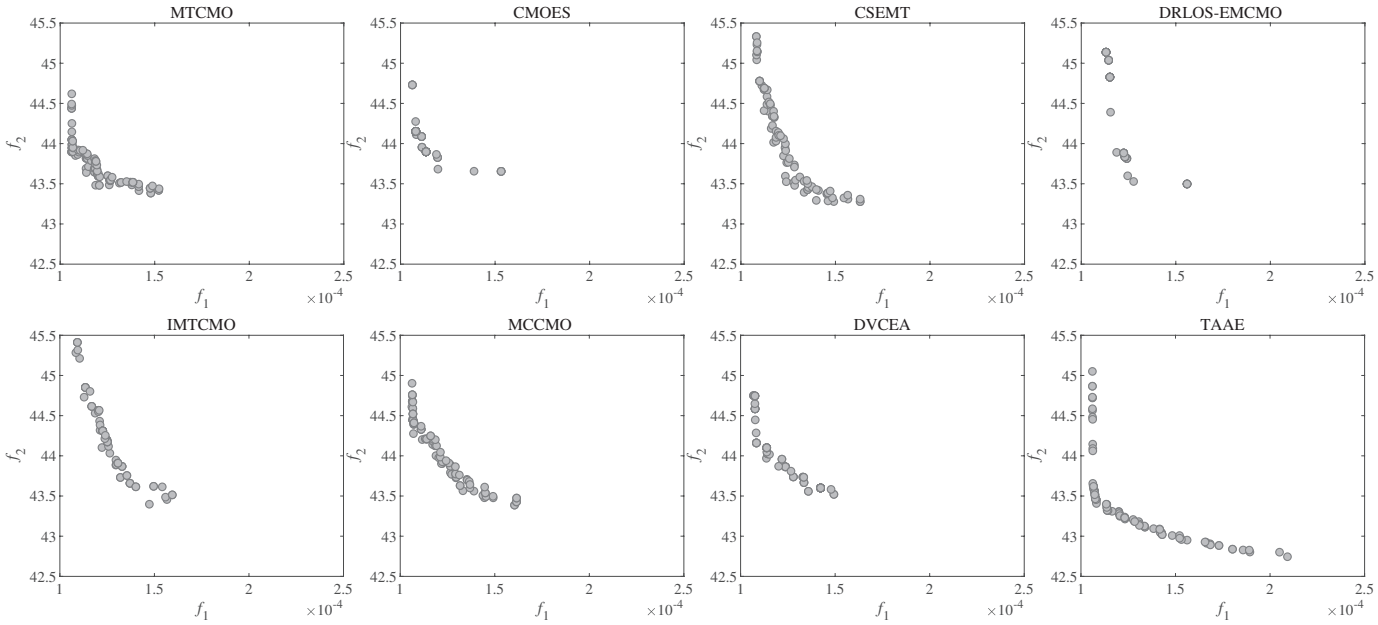


Fig. S26: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS1.

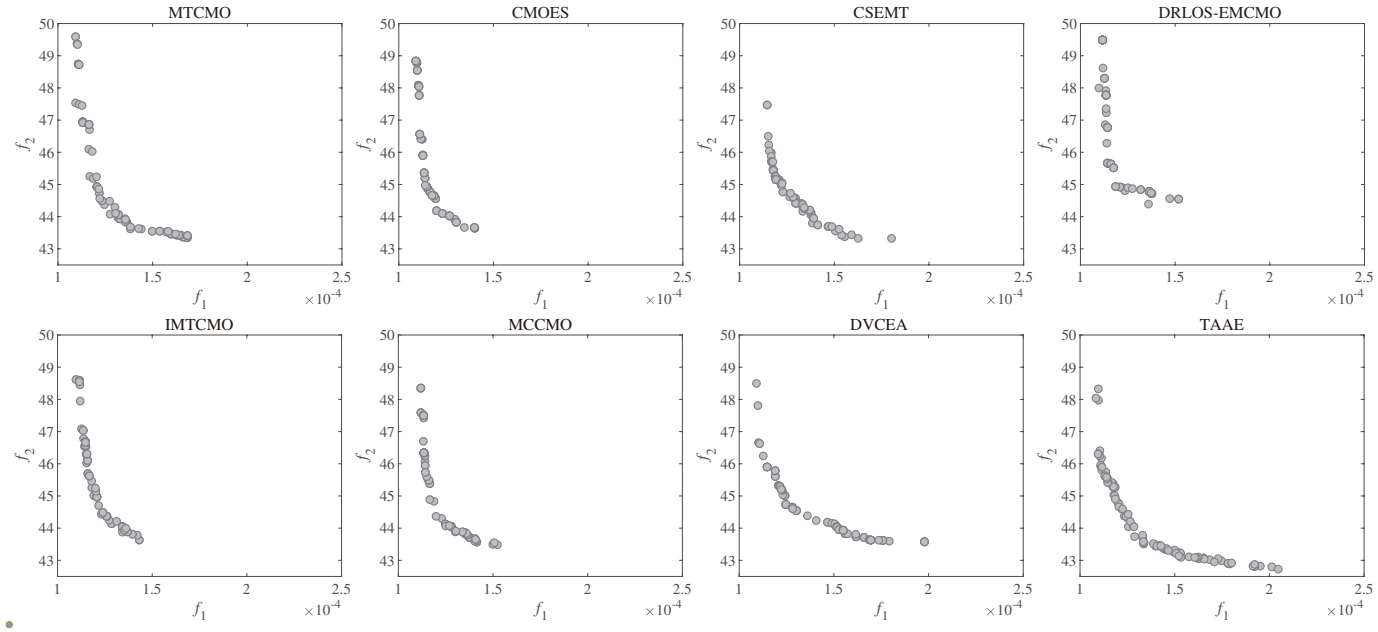


Fig. S27: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS2.

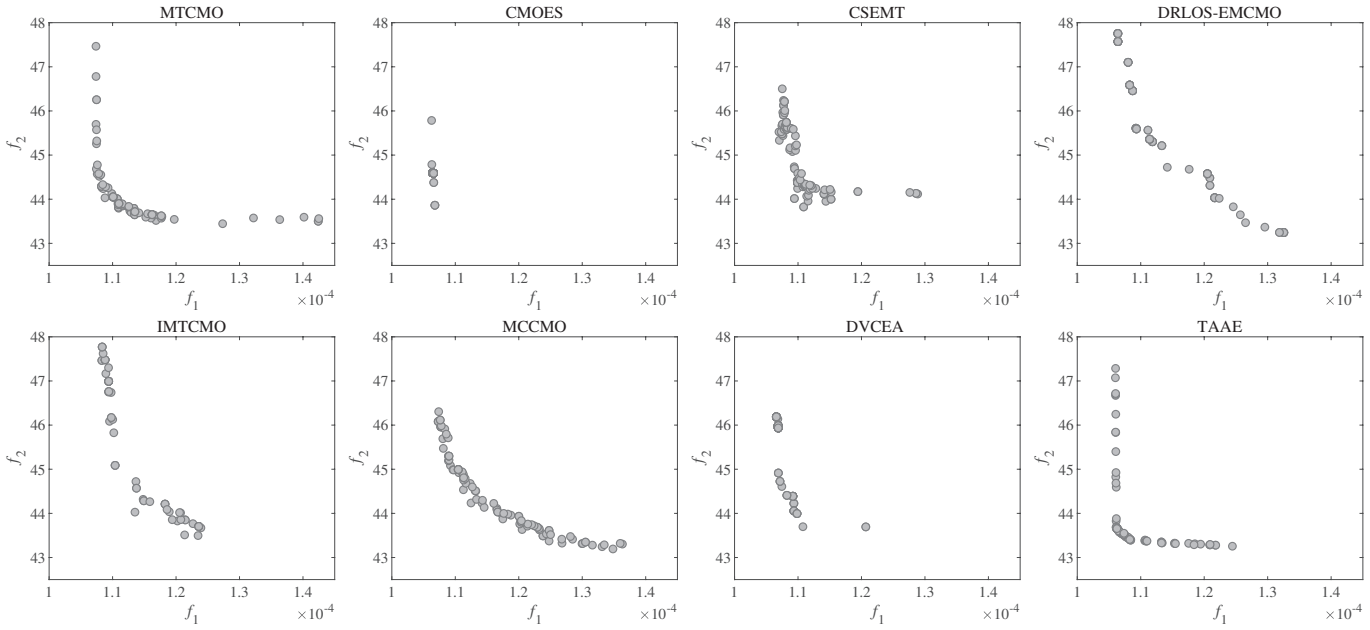


Fig. S28: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS3.

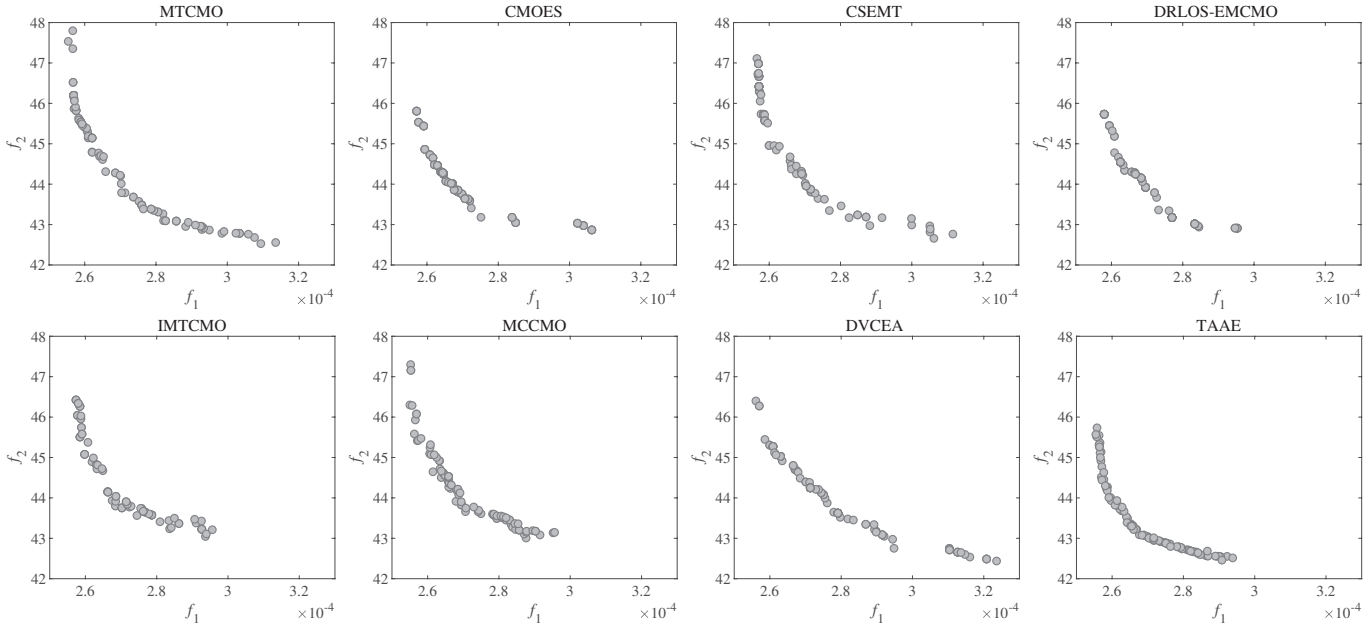


Fig. S29: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS4.

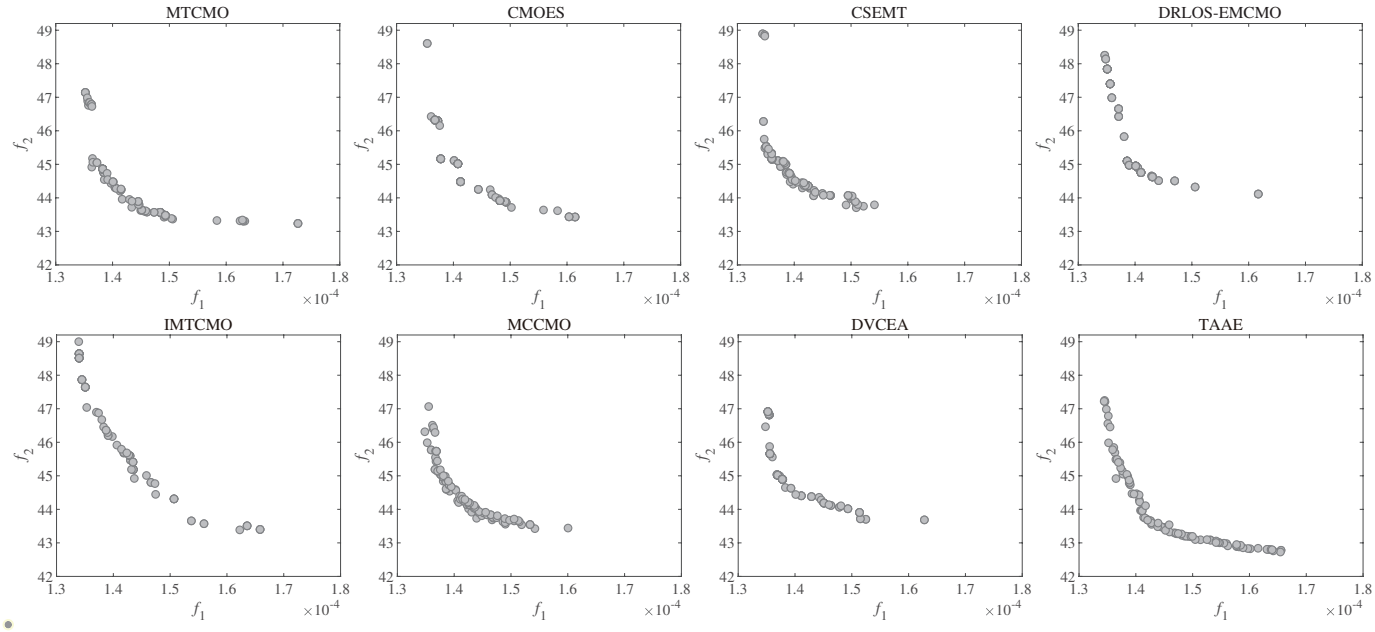


Fig. S30: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS5.

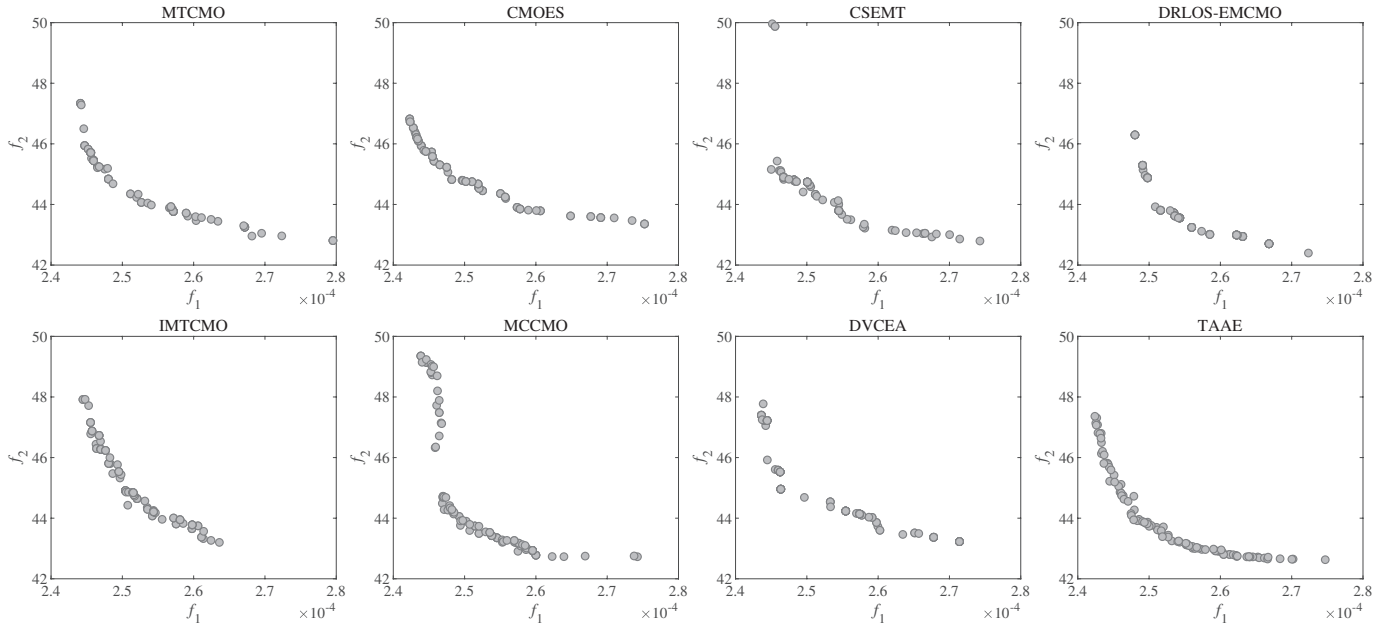


Fig. S31: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS6.

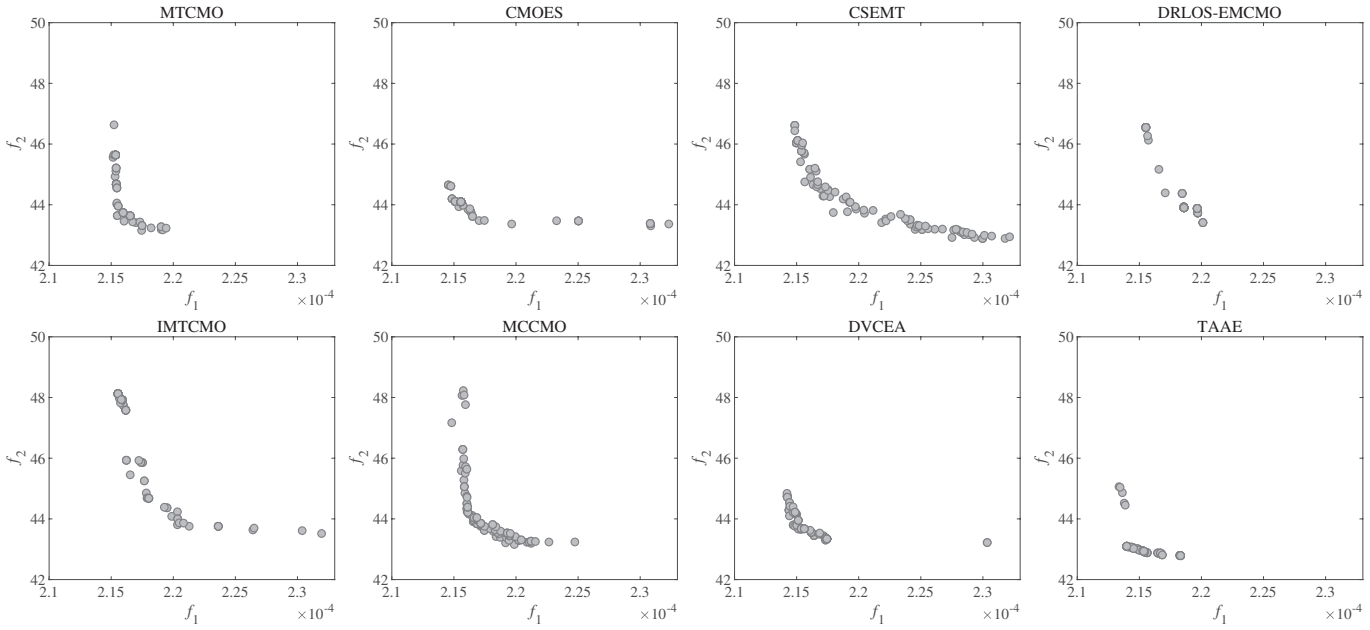


Fig. S32: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS7.

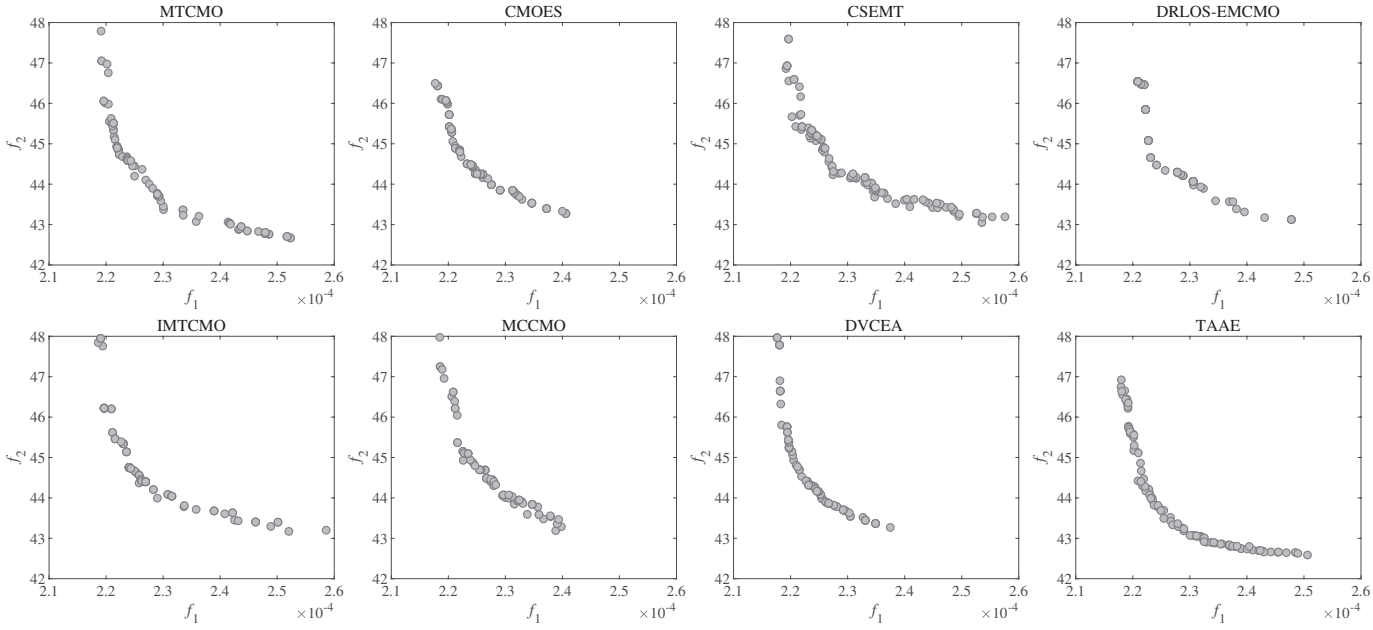


Fig. S33: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C4 test suite under WS8.

TABLE S11: STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C5 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	6.0461e-2 (2.97e-3) —	5.8094e-2 (4.08e-3) —	5.7035e-2 (5.55e-3) —	5.5912e-2 (4.21e-3) —	5.7692e-2 (4.49e-3) —	5.6974e-2 (3.10e-3) —	5.8592e-2 (4.59e-3) —	6.5990e-2 (1.52e-3)
WS2	1.0284e-1 (4.22e-3) —	9.9029e-2 (4.21e-3) —	9.6921e-2 (4.22e-3) —	9.8912e-2 (5.92e-3) —	1.0235e-1 (5.36e-3) —	9.6479e-2 (4.14e-3) —	1.0021e-1 (4.99e-3) —	1.0703e-1 (2.11e-3)
WS3	2.9036e-2 (1.16e-3) —	2.9000e-2 (1.17e-3) —	2.7278e-2 (2.48e-3) —	2.6631e-2 (2.93e-3) —	2.6529e-2 (1.52e-3) —	2.7309e-2 (2.46e-3) —	2.8308e-2 (2.53e-3) —	3.2068e-2 (1.33e-3)
WS4	5.1117e-2 (2.42e-3) —	5.1224e-2 (2.62e-3) —	5.1640e-2 (3.10e-3) —	5.0291e-2 (3.75e-3) —	5.1883e-2 (1.66e-3) —	5.1279e-2 (3.12e-3) —	5.1326e-2 (2.20e-3) —	5.5031e-2 (9.80e-4)
WS5	6.5043e-2 (3.07e-3) —	6.3597e-2 (1.74e-3) —	6.3167e-2 (3.11e-3) —	5.9719e-2 (5.12e-3) —	6.2877e-2 (2.93e-3) —	6.2667e-2 (3.30e-3) —	6.4455e-2 (2.43e-3) —	6.8194e-2 (1.21e-3)
WS6	2.7701e-2 (1.63e-3) —	2.6241e-2 (1.33e-3) —	2.6350e-2 (1.82e-3) —	2.5447e-2 (2.50e-3) —	2.7411e-2 (1.56e-3) —	2.5724e-2 (2.17e-3) —	2.6781e-2 (1.91e-3) —	2.9597e-2 (3.96e-4)
WS7	2.6232e-2 (1.28e-3) —	2.5569e-2 (1.32e-3) —	2.5587e-2 (1.78e-3) —	2.5914e-2 (1.37e-3) —	2.4382e-2 (1.61e-3) —	2.5896e-2 (1.99e-3) —	2.5611e-2 (1.74e-3) —	2.8283e-2 (3.17e-4)
WS8	4.0763e-2 (2.25e-3) —	3.9057e-2 (1.93e-3) —	4.0687e-2 (2.62e-3) —	3.8738e-2 (3.76e-3) —	3.9641e-2 (2.09e-3) —	3.9439e-2 (2.81e-3) —	4.0209e-2 (2.07e-3) —	4.3982e-2 (1.11e-3)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

TABLE S12: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C5 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	1.7056e-1 (1.19e-1) —	3.3339e-1 (2.61e-1) —	3.5838e-1 (2.70e-1) —	3.3166e-1 (3.20e-1) —	2.8954e-1 (2.55e-1) —	3.4077e-1 (2.25e-1) —	3.0434e-1 (2.41e-1) —	5.6679e-2 (2.41e-2)
WS2	3.0219e-1 (1.18e-1) —	6.1567e-1 (2.17e-1) —	6.0639e-1 (2.24e-1) —	5.3534e-1 (3.44e-1) —	3.3304e-1 (2.06e-1) $\approx$	6.1558e-1 (1.91e-1) —	5.0320e-1 (2.17e-1) —	2.1584e-1 (1.22e-1)
WS3	1.7634e-1 (1.36e-1) $\approx$	2.5421e-1 (1.36e-1) —	3.6302e-1 (3.19e-1) —	3.3418e-1 (5.10e-1) $\approx$	1.4600e-1 (1.86e-1) $\approx$	3.6750e-1 (3.07e-1) $\approx$	2.6341e-1 (2.28e-1) $\approx$	1.2699e-1 (4.85e-2)
WS4	2.8902e-1 (1.07e-1) —	3.2777e-1 (1.56e-1) —	3.5583e-1 (1.67e-1) —	3.8871e-1 (3.18e-1) —	2.0737e-1 (1.15e-1) $\approx$	3.2680e-1 (1.86e-1) —	2.7218e-1 (1.03e-1) —	1.3299e-1 (4.41e-2)
WS5	3.2497e-1 (1.45e-1) —	4.3314e-1 (1.55e-1) —	3.6058e-1 (1.82e-1) —	5.0691e-1 (3.56e-1) —	3.3829e-1 (1.18e-1) —	4.4622e-1 (2.58e-1) —	3.5257e-1 (1.73e-1) —	1.9936e-1 (8.29e-2)
WS6	2.7508e-1 (1.94e-1) —	4.6466e-1 (2.24e-1) —	3.4473e-1 (2.03e-1) —	3.7134e-1 (3.49e-1) —	2.1860e-1 (1.90e-1) —	5.7115e-1 (2.55e-1) —	3.3950e-1 (2.06e-1) —	9.4344e-2 (3.15e-2)
WS7	2.6405e-1 (2.11e-1) —	4.4691e-1 (2.63e-1) —	4.2041e-1 (3.50e-1) —	2.1466e-1 (2.30e-1) —	4.5634e-1 (3.38e-1) —	3.3711e-1 (3.66e-1) $\approx$	4.2173e-1 (3.39e-1) —	6.9378e-2 (1.64e-2)
WS8	2.6262e-1 (1.71e-1) —	4.2128e-1 (1.84e-1) —	2.7686e-1 (2.19e-1) —	3.4726e-1 (4.05e-1) —	2.5762e-1 (1.99e-1) —	3.6157e-1 (2.42e-1) —	2.7762e-1 (1.38e-1) —	9.4059e-2 (6.47e-2)
$+/-/\approx$	0/7/1	0/8/0	0/8/0	0/7/1	0/5/3	0/6/2	0/7/1	

TABLE S13: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C5 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	1.5532e-1 (1.28e-1) —	3.2126e-1 (2.68e-1) —	3.4122e-1 (2.85e-1) —	3.0478e-1 (3.36e-1) —	2.6412e-1 (2.71e-1) —	3.2686e-1 (2.36e-1) —	2.8687e-1 (2.55e-1) —	2.7530e-2 (1.79e-2)
WS2	2.1352e-1 (1.45e-1) $\approx$	5.3511e-1 (2.55e-1) —	5.2951e-1 (2.34e-1) —	3.8066e-1 (3.81e-1) —	2.6643e-1 (2.13e-1) —	5.3609e-1 (2.18e-1) —	4.2589e-1 (2.32e-1) —	1.2691e-1 (1.13e-1)
WS3	1.5403e-1 (1.44e-1) —	2.2408e-1 (1.47e-1) —	3.5033e-1 (3.25e-1) —	3.0942e-1 (5.22e-1) —	1.1033e-1 (1.96e-1) —	3.4731e-1 (3.22e-1) —	2.4301e-1 (2.37e-1) —	1.0618e-2 (1.01e-2)
WS4	2.0161e-1 (1.15e-1) —	2.5300e-1 (1.76e-1) —	2.3074e-1 (2.16e-1) —	2.3782e-1 (3.53e-1) —	1.1519e-1 (1.02e-1) —	2.0942e-1 (2.29e-1) —	1.9340e-1 (1.21e-1) —	4.7334e-2 (2.64e-2)
WS5	2.3456e-1 (1.58e-1) —	3.5063e-1 (1.71e-1) —	2.7790e-1 (1.97e-1) —	4.2436e-1 (3.92e-1) —	2.6543e-1 (1.35e-1) —	3.7635e-1 (2.73e-1) —	2.8107e-1 (1.82e-1) —	1.1260e-1 (6.66e-2)
WS6	2.3917e-1 (2.07e-1) —	4.3923e-1 (2.29e-1) —	3.1326e-1 (2.10e-1) —	3.1977e-1 (3.68e-1) —	1.7863e-1 (1.99e-1) $\approx$	5.4770e-1 (2.68e-1) —	3.1263e-1 (2.15e-1) —	6.1644e-2 (3.01e-2)
WS7	2.4722e-1 (2.23e-1) $\approx$	4.3706e-1 (2.68e-1) —	4.0582e-1 (3.61e-1) $\approx$	1.8462e-1 (2.45e-1) $\approx$	4.3458e-1 (3.58e-1) —	3.1090e-1 (3.83e-1) $\approx$	4.0373e-1 (3.55e-1) —	4.5554e-2 (1.79e-2)
WS8	2.1174e-1 (1.87e-1) —	3.8970e-1 (1.88e-1) —	2.2647e-1 (2.40e-1) —	2.8866e-1 (4.22e-1) —	2.1227e-1 (2.06e-1) —	3.2860e-1 (2.56e-1) —	2.3473e-1 (1.49e-1) —	4.6686e-2 (6.76e-2)
$+/-/\approx$	0/6/2	0/8/0	0/7/1	0/7/1	0/7/1	0/7/1	0/8/0	

TABLE S14: STATISTICAL RESULTS OF HV OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C6 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	6.9151e-2 (3.65e-3) —	6.8430e-2 (6.14e-3) —	6.6914e-2 (5.08e-3) —	6.6124e-2 (8.52e-3) —	7.1543e-2 (5.28e-3) —	6.9876e-2 (6.53e-3) —	7.0248e-2 (4.03e-3) —	7.8472e-2 (7.12e-3)
WS2	1.0461e-1 (5.07e-3) —	1.0250e-1 (7.48e-3) —	1.0264e-1 (5.51e-3) —	9.8483e-2 (1.24e-2) —	1.0716e-1 (5.99e-3) —	1.0109e-1 (5.12e-3) —	1.0549e-1 (5.57e-3) —	1.1194e-1 (3.85e-3)
WS3	1.9133e-2 (1.60e-3) —	1.9845e-2 (2.06e-3) —	1.8329e-2 (2.16e-3) —	1.6631e-2 (4.05e-3) —	1.6720e-2 (2.88e-3) —	1.7539e-2 (2.94e-3) —	1.8752e-2 (2.47e-3) —	2.3750e-2 (1.66e-3)
WS4	3.7861e-2 (1.86e-3) —	3.9159e-2 (2.60e-3) —	3.7317e-2 (2.66e-3) —	3.5898e-2 (4.71e-3) —	3.8134e-2 (2.93e-3) —	3.7762e-2 (2.94e-3) —	3.9033e-2 (2.65e-3) —	4.2155e-2 (1.41e-3)
WS5	4.2464e-2 (2.51e-3) —	4.1559e-2 (1.43e-3) —	3.9694e-2 (2.48e-3) —	3.6368e-2 (5.70e-3) —	3.9578e-2 (3.35e-3) —	3.9436e-2 (2.95e-3) —	4.1671e-2 (1.85e-3) —	4.5902e-2 (1.69e-3)
WS6	3.5474e-2 (1.87e-3) —	3.4386e-2 (2.84e-3) —	3.3709e-2 (3.09e-3) —	3.2113e-2 (4.72e-3) —	3.6095e-2 (2.69e-3) —	3.3632e-2 (2.76e-3) —	3.5152e-2 (2.26e-3) —	3.9317e-2 (1.07e-3)
WS7	2.5030e-2 (1.84e-3) —	2.5176e-2 (2.30e-3) —	2.4838e-2 (1.69e-3) —	2.2952e-2 (3.29e-3) —	2.5300e-2 (2.26e-3) —	2.4089e-2 (1.69e-3) —	2.5365e-2 (1.83e-3) —	2.7716e-2 (1.25e-3)
WS8	3.5070e-2 (1.68e-3) —	3.3912e-2 (2.48e-3) —	3.2192e-2 (2.04e-3) —	3.4150e-2 (3.87e-3) —	3.3799e-2 (3.50e-3) —	3.4854e-2 (3.42e-3) —	3.4565e-2 (2.52e-3) —	3.8116e-2 (2.23e-3)
$+/-/\approx$	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	0/8/0	

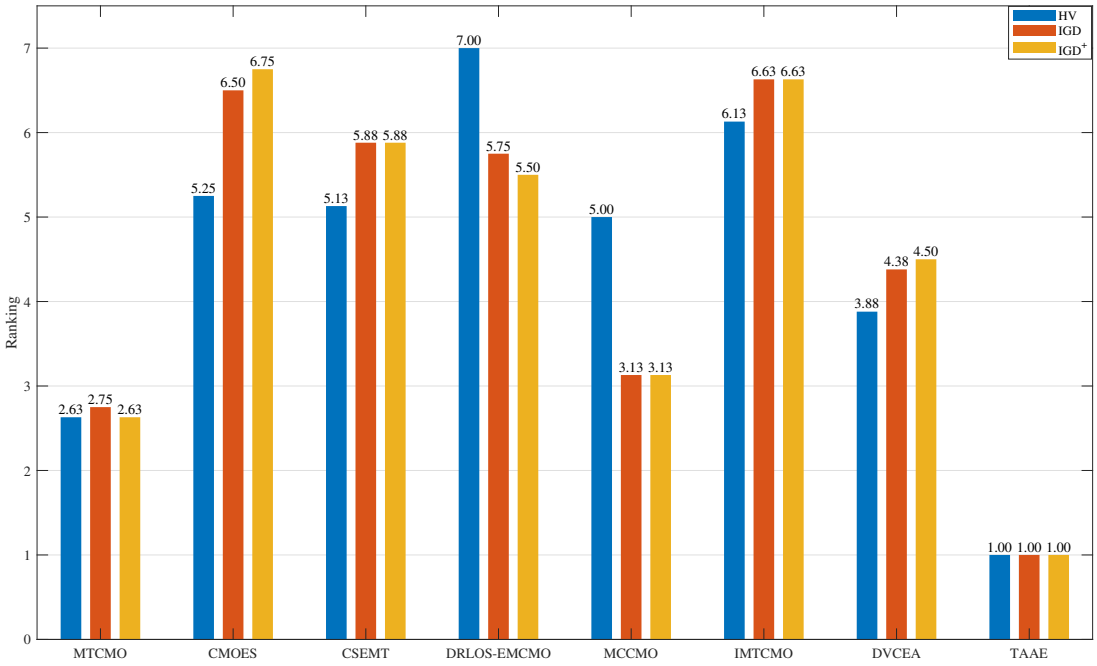


Fig. S34: Friedman test results for TAAE and other compared algorithms on the C5 test suite.

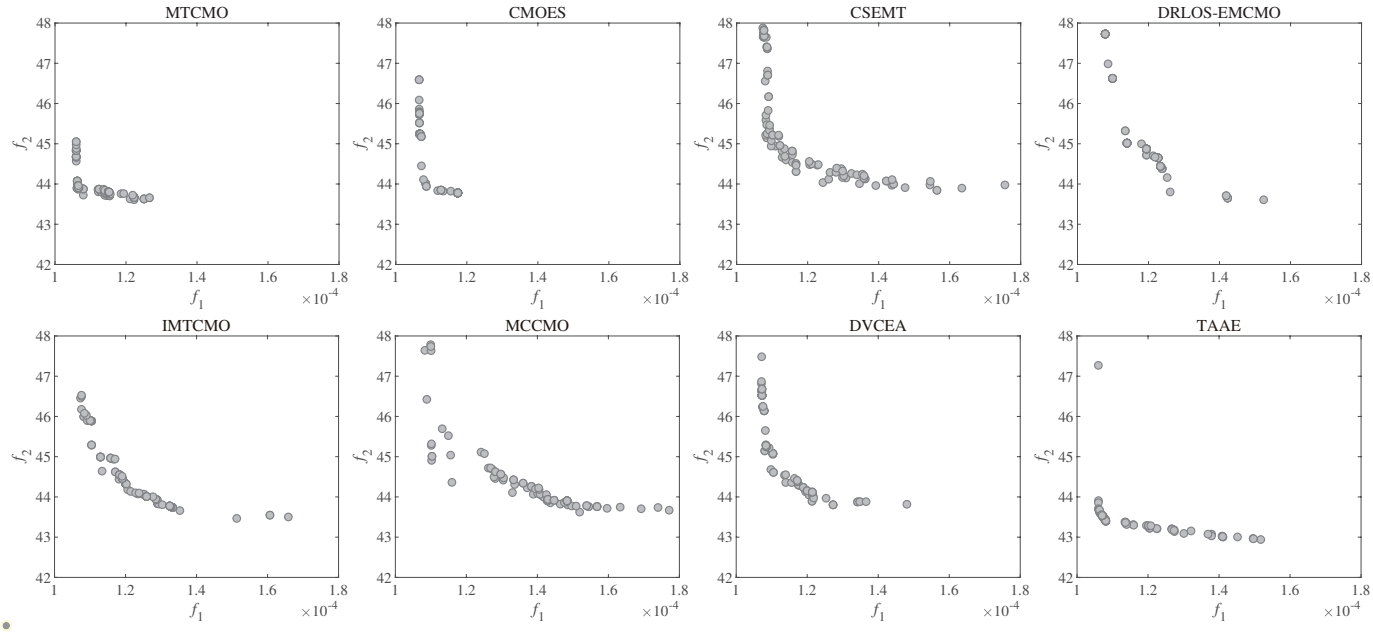


Fig. S35: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS1.

TABLE S15: STATISTICAL RESULTS OF IGD OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C6 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	7.0040e-1 (2.75e-1) —	8.7049e-1 (4.84e-1) —	8.7214e-1 (4.05e-1) —	7.7030e-1 (7.45e-1) —	5.0337e-1 (3.78e-1) —	7.1562e-1 (4.63e-1) —	6.1733e-1 (2.66e-1) —	2.8571e-1 (4.16e-1)
WS2	5.0670e-1 (1.93e-1) —	7.1223e-1 (3.24e-1) —	6.1196e-1 (1.86e-1) —	8.3372e-1 (5.88e-1) —	3.9955e-1 (2.27e-1) ≈	6.1545e-1 (1.66e-1) —	5.2462e-1 (1.70e-1) —	2.8857e-1 (1.46e-1)
WS3	5.7810e-1 (2.35e-1) —	5.9440e-1 (3.89e-1) —	6.9934e-1 (3.09e-1) —	6.2409e-1 (6.24e-1) —	6.6348e-1 (5.02e-1) —	8.7958e-1 (5.17e-1) —	5.0625e-1 (3.43e-1) —	7.7573e-2 (1.46e-1)
WS4	4.3859e-1 (1.71e-1) —	4.2052e-1 (2.34e-1) —	5.2860e-1 (2.55e-1) —	5.1627e-1 (5.13e-1) —	3.4063e-1 (2.35e-1) —	4.7866e-1 (3.29e-1) —	3.5582e-1 (2.16e-1) —	1.2603e-1 (9.88e-2)
WS5	3.3642e-1 (2.05e-1) —	5.1056e-1 (1.95e-1) —	6.1037e-1 (2.95e-1) —	7.3680e-1 (5.84e-1) —	5.2314e-1 (3.02e-1) —	6.6287e-1 (1.98e-1) —	4.0385e-1 (1.69e-1) —	1.2378e-1 (5.73e-2)
WS6	6.0977e-1 (2.12e-1) —	8.6312e-1 (3.71e-1) —	8.2628e-1 (3.29e-1) —	9.4385e-1 (5.64e-1) —	5.0278e-1 (2.74e-1) —	8.6604e-1 (3.00e-1) —	6.8867e-1 (2.37e-1) —	2.5454e-1 (9.29e-2)
WS7	5.7307e-1 (2.99e-1) —	6.0299e-1 (5.25e-1) —	5.7298e-1 (3.55e-1) —	8.4711e-1 (6.96e-1) —	3.6893e-1 (4.31e-1) ≈	7.5855e-1 (3.30e-1) —	5.3555e-1 (3.44e-1) —	1.9090e-1 (1.40e-1)
WS8	4.4925e-1 (1.53e-1) —	6.5611e-1 (2.70e-1) —	7.5978e-1 (3.46e-1) —	4.3860e-1 (4.24e-1) ≈	4.7704e-1 (3.60e-1) —	4.8555e-1 (3.99e-1) ≈	4.5255e-1 (2.38e-1) —	1.8884e-1 (1.55e-1)
+/ − / ≈	0/8/0	0/8/0	0/8/0	0/7/1	0/6/2	0/7/1	0/8/0	

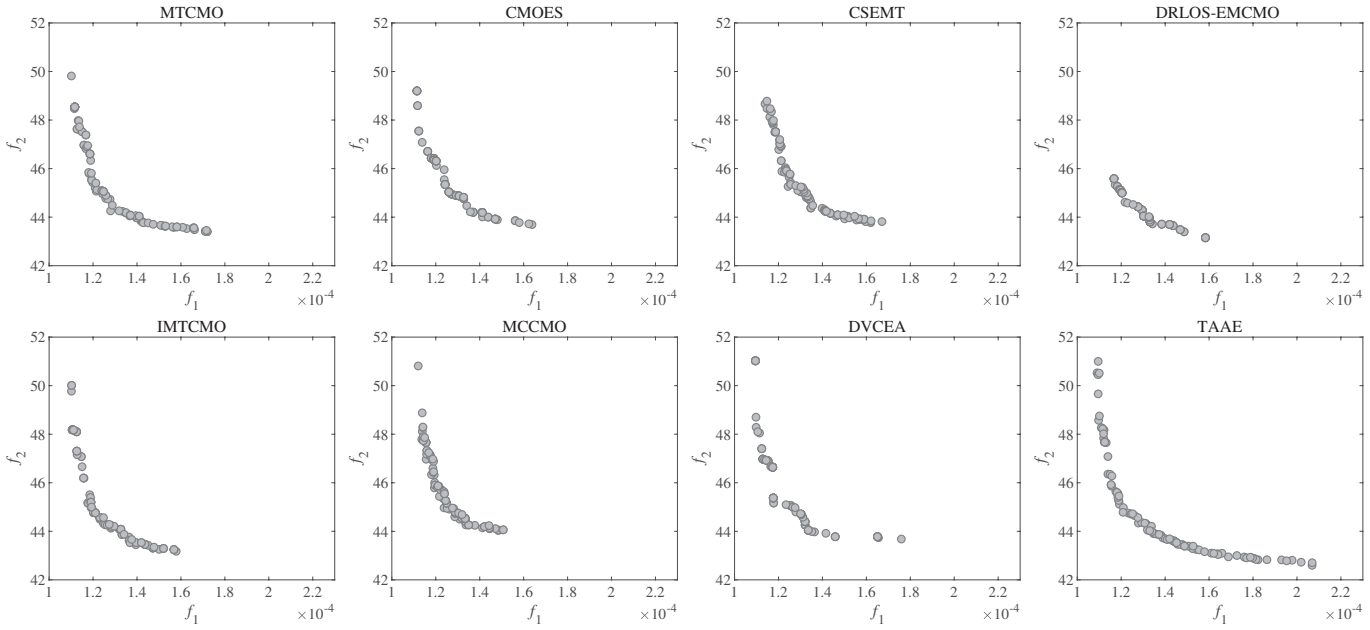


Fig. S36: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS2.

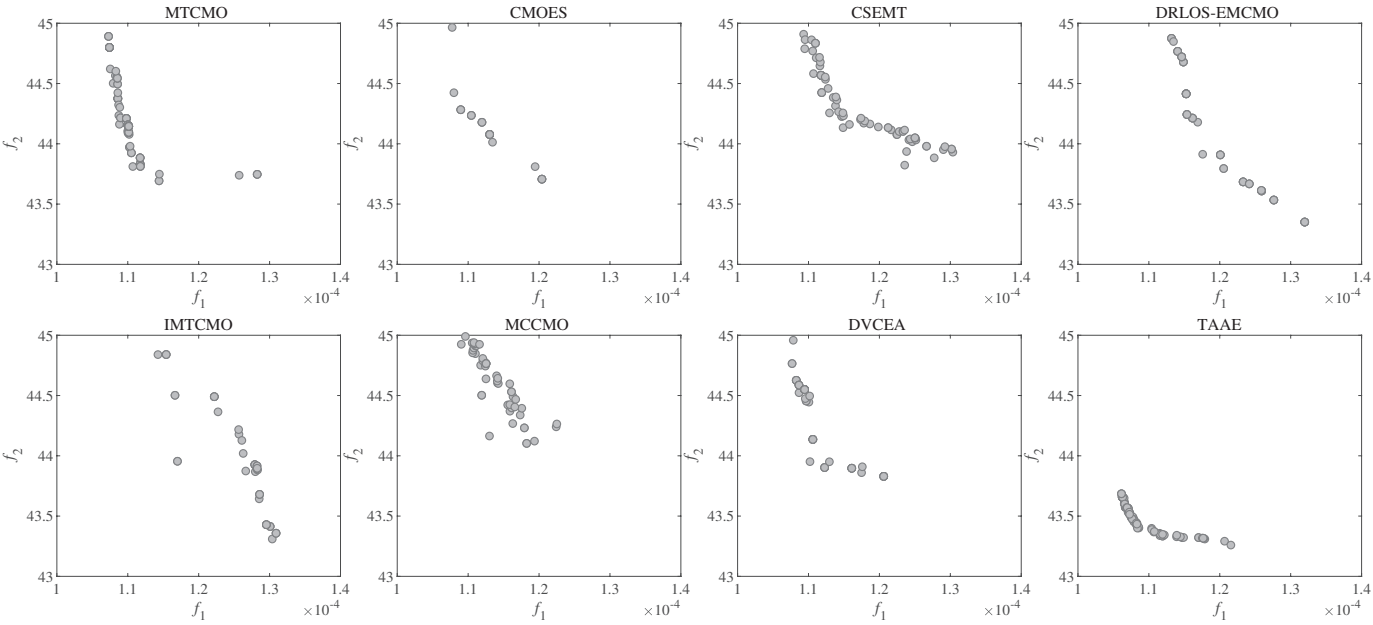


Fig. S37: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS3.

TABLE S16: STATISTICAL RESULTS OF IGD$^+$ OBTAINED BY MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, AND TAAE ON THE C6 TEST SUITE.

Problem	MTCMO	CMOES	CSEMT	DRLOS-EMCMO	IMTCMO	MCCMO	DVCEA	TAAE
WS1	6.8597e-1 (2.83e-1) −	8.5604e-1 (4.96e-1) −	8.6171e-1 (4.11e-1) −	7.4977e-1 (7.57e-1) −	4.9000e-1 (3.87e-1) −	7.0766e-1 (4.67e-1) −	6.0728e-1 (2.70e-1) −	2.4099e-1 (4.32e-1)
WS2	4.3466e-1 (2.07e-1) −	6.3132e-1 (3.68e-1) −	5.1764e-1 (2.18e-1) −	6.9305e-1 (6.72e-1) −	3.1368e-1 (2.44e-1) ≈	5.3534e-1 (2.07e-1) −	4.4146e-1 (2.22e-1) −	1.9993e-1 (1.41e-1)
WS3	5.7592e-1 (2.40e-1) −	5.9091e-1 (3.93e-1) −	6.9777e-1 (3.13e-1) −	6.1787e-1 (6.30e-1) −	6.5590e-1 (5.11e-1) −	8.7769e-1 (5.20e-1) −	5.0185e-1 (3.49e-1) −	6.8638e-2 (1.49e-1)
WS4	4.2064e-1 (1.77e-1) −	3.9412e-1 (2.47e-1) −	5.0625e-1 (2.69e-1) −	4.7559e-1 (5.35e-1) −	3.1566e-1 (2.45e-1) −	4.5507e-1 (3.39e-1) −	3.2708e-1 (2.34e-1) −	8.7432e-2 (1.07e-1)
WS5	2.9563e-1 (2.22e-1) −	4.7550e-1 (2.02e-1) −	5.7251e-1 (3.10e-1) −	6.9695e-1 (6.01e-1) −	4.9127e-1 (3.14e-1) −	6.3297e-1 (2.11e-1) −	3.7715e-1 (1.76e-1) −	8.7194e-2 (6.23e-2)
WS6	5.8534e-1 (2.19e-1) −	8.3730e-1 (3.84e-1) −	7.9478e-1 (3.56e-1) −	8.8879e-1 (5.98e-1) −	4.5582e-1 (2.84e-1) −	8.4643e-1 (3.06e-1) −	6.5782e-1 (2.52e-1) −	2.2814e-1 (9.57e-2)
WS7	5.5905e-1 (3.14e-1) −	5.8539e-1 (5.36e-1) −	5.6332e-1 (3.61e-1) −	8.3718e-1 (7.02e-1) −	3.4720e-1 (4.43e-1) ≈	7.4978e-1 (3.34e-1) −	5.2478e-1 (3.53e-1) −	1.5214e-1 (1.58e-1)
WS8	4.3297e-1 (1.57e-1) −	6.4319e-1 (2.77e-1) −	7.4920e-1 (3.53e-1) −	4.0466e-1 (4.41e-1) ≈	4.5481e-1 (3.73e-1) −	4.6452e-1 (4.13e-1) ≈	4.3748e-1 (2.45e-1) −	1.6255e-1 (1.61e-1)
+/ − / ≈	0/8/0	0/8/0	0/8/0	0/7/1	0/6/2	0/7/1	0/8/0	

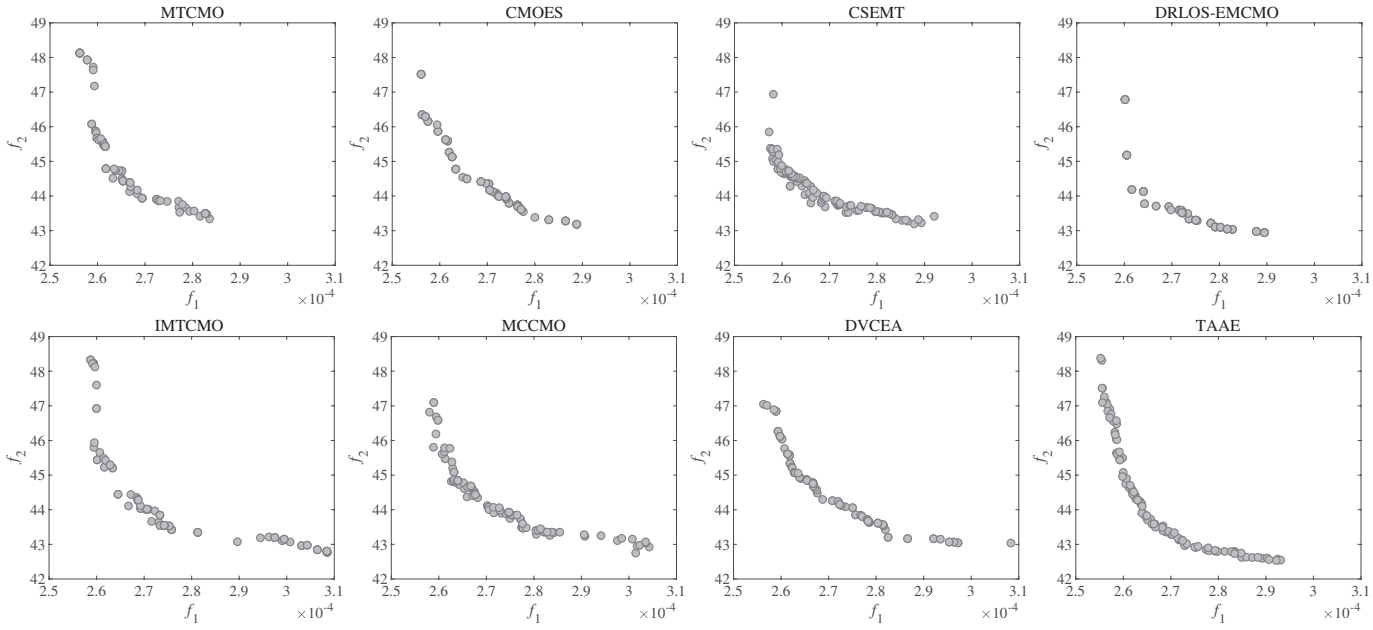


Fig. S38: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS4.

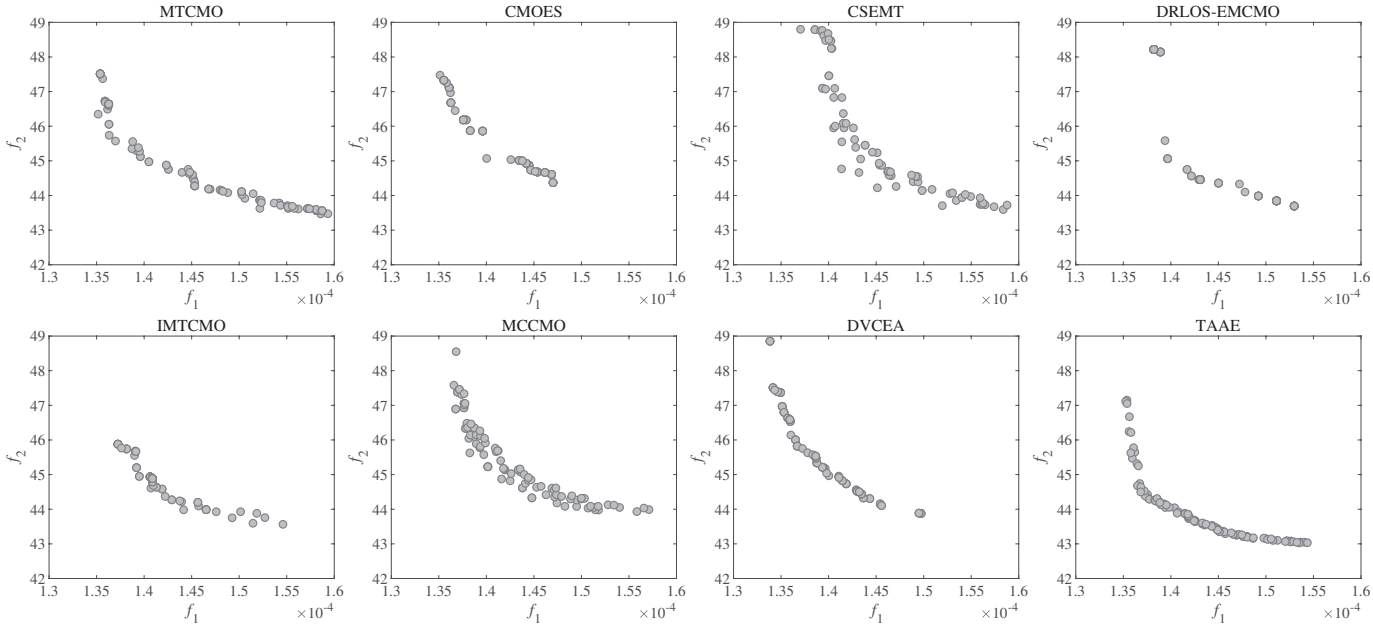


Fig. S39: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS5.

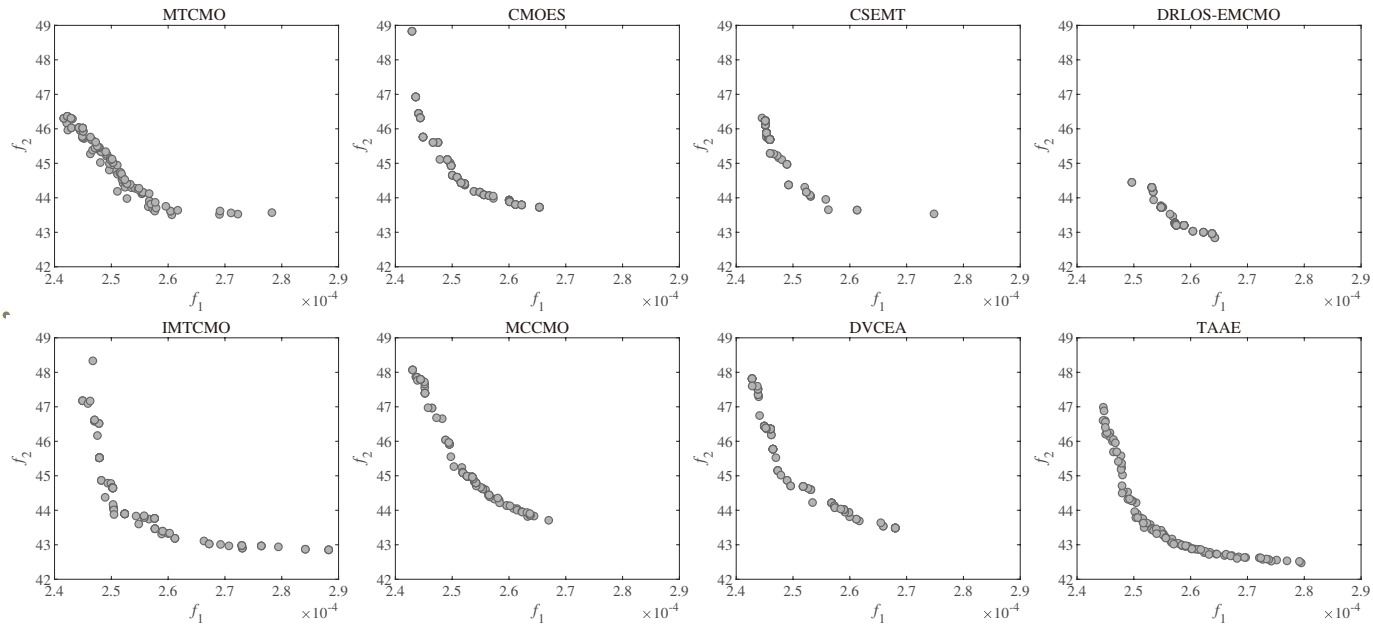


Fig. S40: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS6.

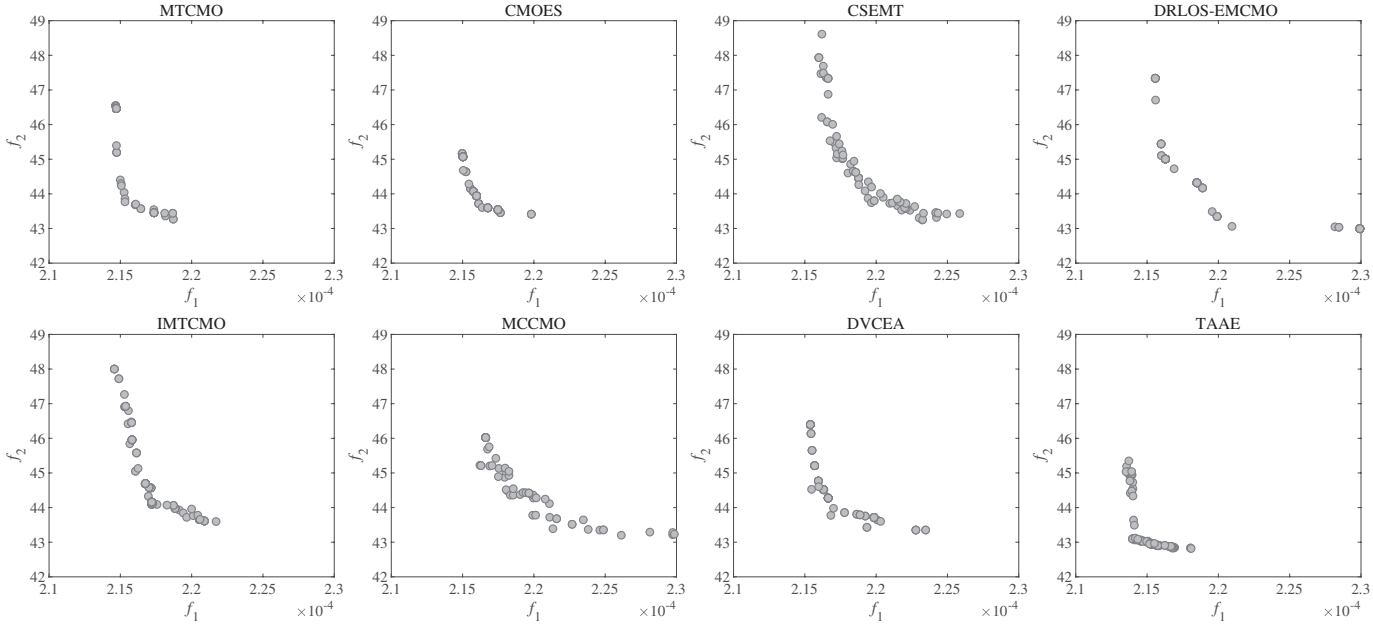


Fig. S41: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS7.

TABLE S17: STATISTICAL RESULTS OF HV OBTAINED BY TAAE_Binary, TAAE_Integer, AND ORIGINAL TAAE.

Problem	TAAE_Binary	TAAE_Integer	TAAE
WS1	6.6767e-2 (1.21e-2) −	1.0005e-1 (1.52e-3) −	1.0497e-1 (1.66e-3)
WS2	4.9298e-2 (6.07e-3) −	7.7544e-2 (7.94e-4) −	8.1066e-2 (1.04e-3)
WS3	1.8834e-2 (3.06e-3) −	3.6318e-2 (3.08e-4) −	3.7994e-2 (3.04e-4)
WS4	1.9143e-2 (3.53e-3) −	3.2266e-2 (3.15e-4) −	3.3284e-2 (4.72e-4)
WS5	1.7595e-2 (2.40e-3) −	3.3730e-2 (5.12e-4) −	3.4989e-2 (3.86e-4)
WS6	1.8003e-2 (2.43e-3) −	2.8231e-2 (2.93e-4) −	2.9362e-2 (5.17e-4)
WS7	7.0354e-3 (2.14e-3) −	1.5500e-2 (1.86e-4) −	1.6161e-2 (3.47e-4)
WS8	3.4299e-2 (3.86e-3) −	4.8971e-2 (4.65e-4) −	4.9669e-2 (5.76e-4)
$+/-/\approx$	0/8/0	0/8/0	

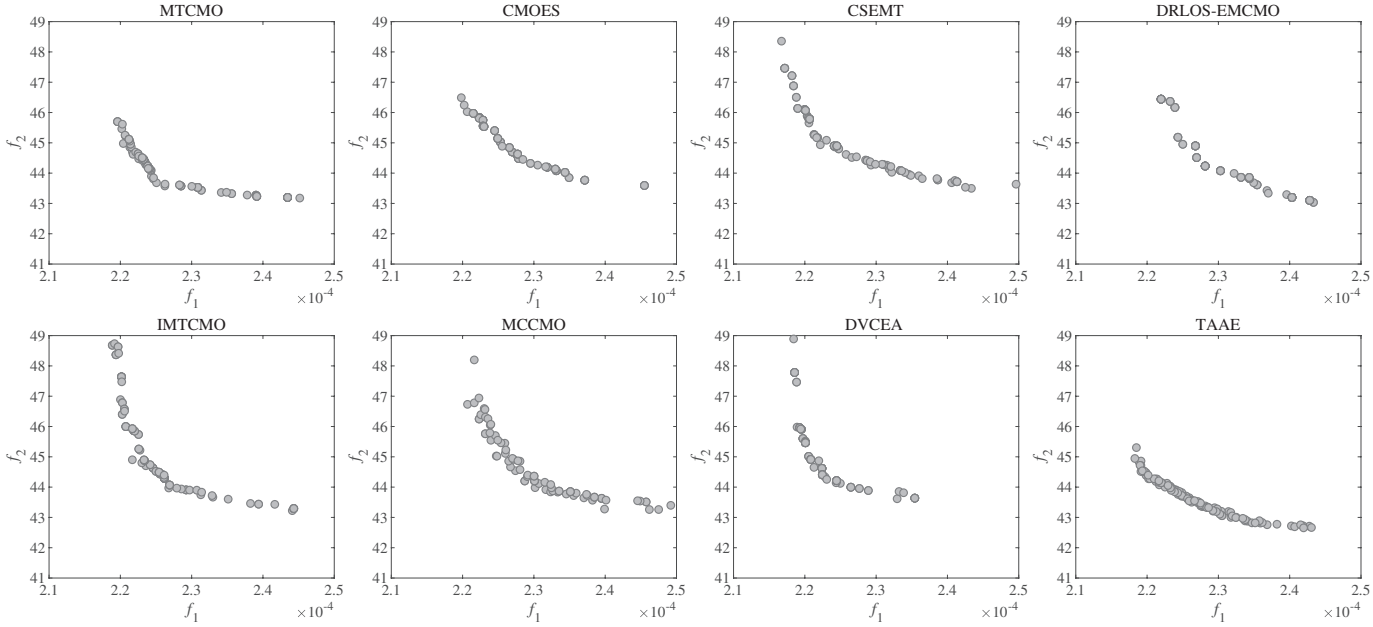


Fig. S42: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C5 test suite under WS8.

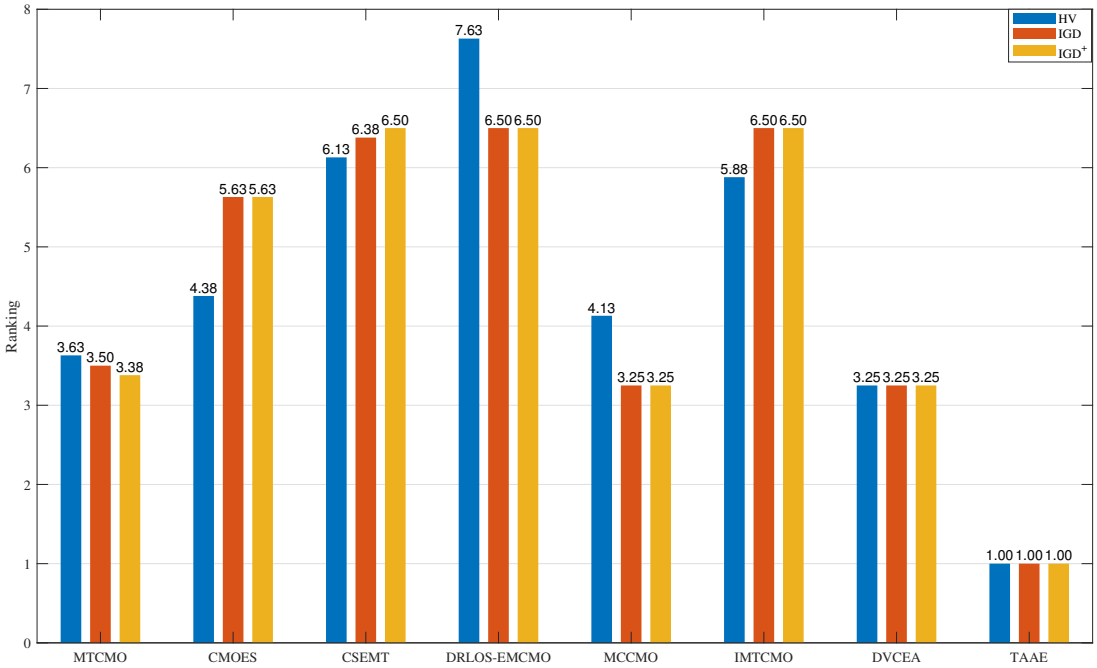


Fig. S43: Friedman test results for TAAE and other compared algorithms on the C6 test suite.

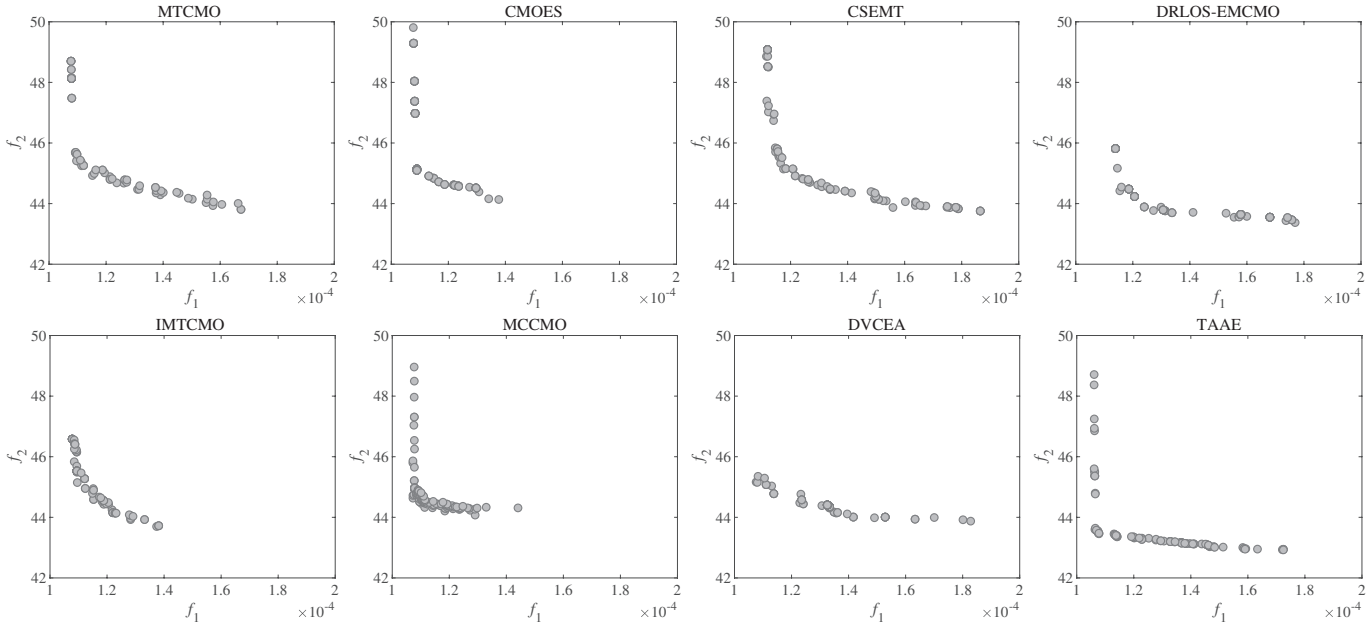


Fig. S44: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS1.

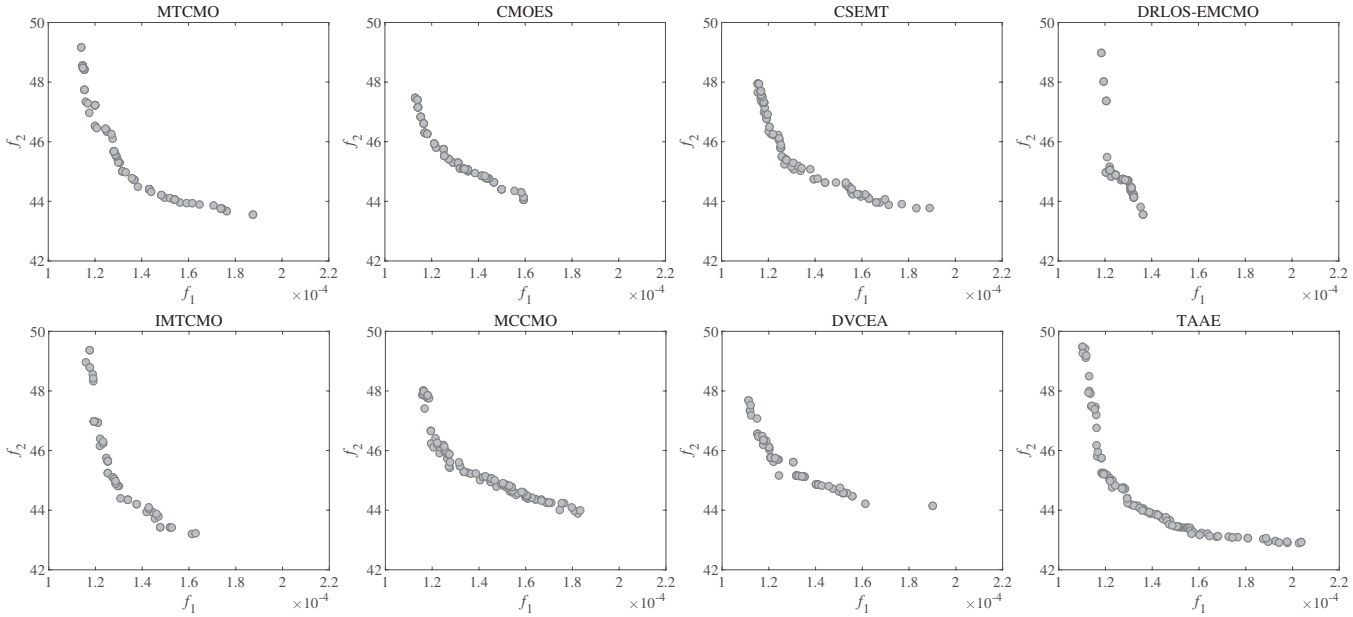


Fig. S45: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS2.

TABLE S18: STATISTICAL RESULTS OF HV OBTAINED BY TAAE_1, TAAE_2, AND ORIGINAL TAAE.

Problem	TAAE_1	TAAE_2	TAAE
WS1	1.0047e-1 (2.40e-3) −	9.2988e-2 (1.46e-3) −	1.0497e-1 (1.66e-3)
WS2	7.6393e-2 (1.81e-3) −	7.2363e-2 (9.27e-4) −	8.1066e-2 (1.04e-3)
WS3	3.4670e-2 (1.21e-3) −	3.4066e-2 (5.48e-4) −	3.7994e-2 (3.04e-4)
WS4	3.1136e-2 (1.08e-3) −	3.0371e-2 (4.27e-4) −	3.3284e-2 (4.72e-4)
WS5	3.2223e-2 (8.43e-4) −	3.1245e-2 (5.65e-4) −	3.4989e-2 (3.86e-4)
WS6	2.7018e-2 (1.46e-3) −	2.6398e-2 (4.61e-4) −	2.9362e-2 (5.17e-4)
WS7	1.4927e-2 (8.17e-4) −	1.4356e-2 (2.43e-4) −	1.6161e-2 (3.47e-4)
WS8	4.7033e-2 (2.35e-3) −	4.6475e-2 (3.61e-4) −	4.9669e-2 (5.76e-4)
$+/-/\approx$	0/8/0	0/8/0	

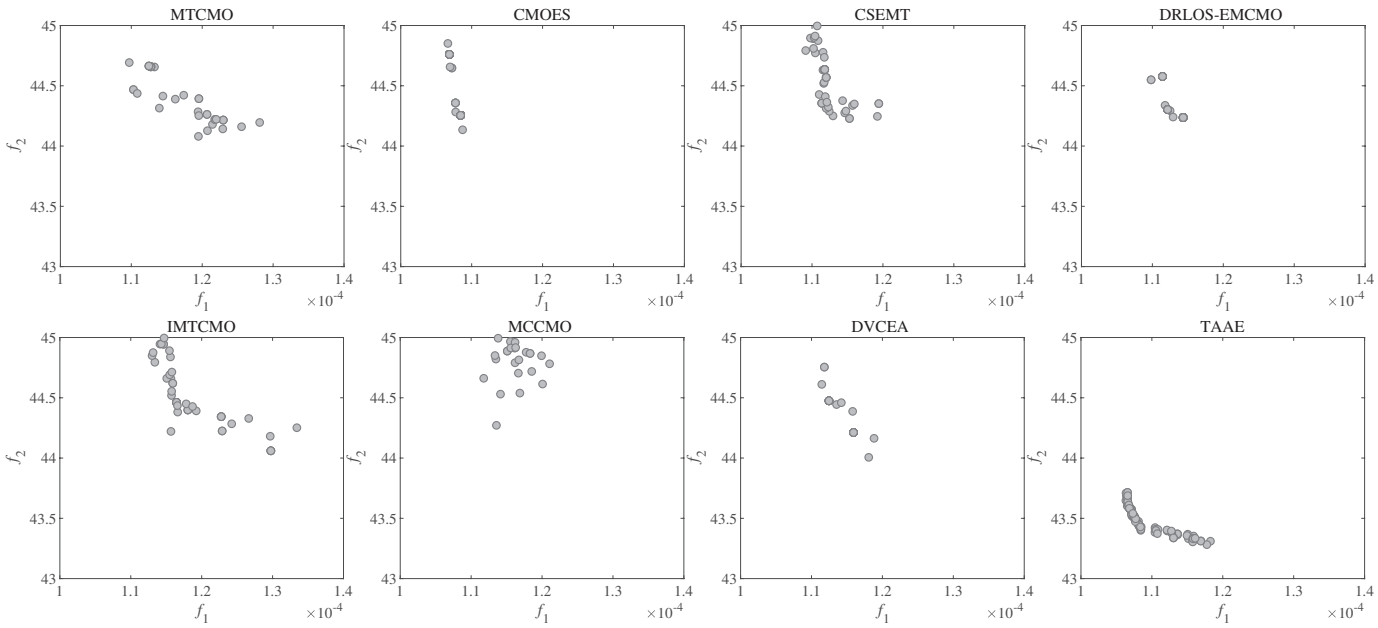


Fig. S46: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS3.

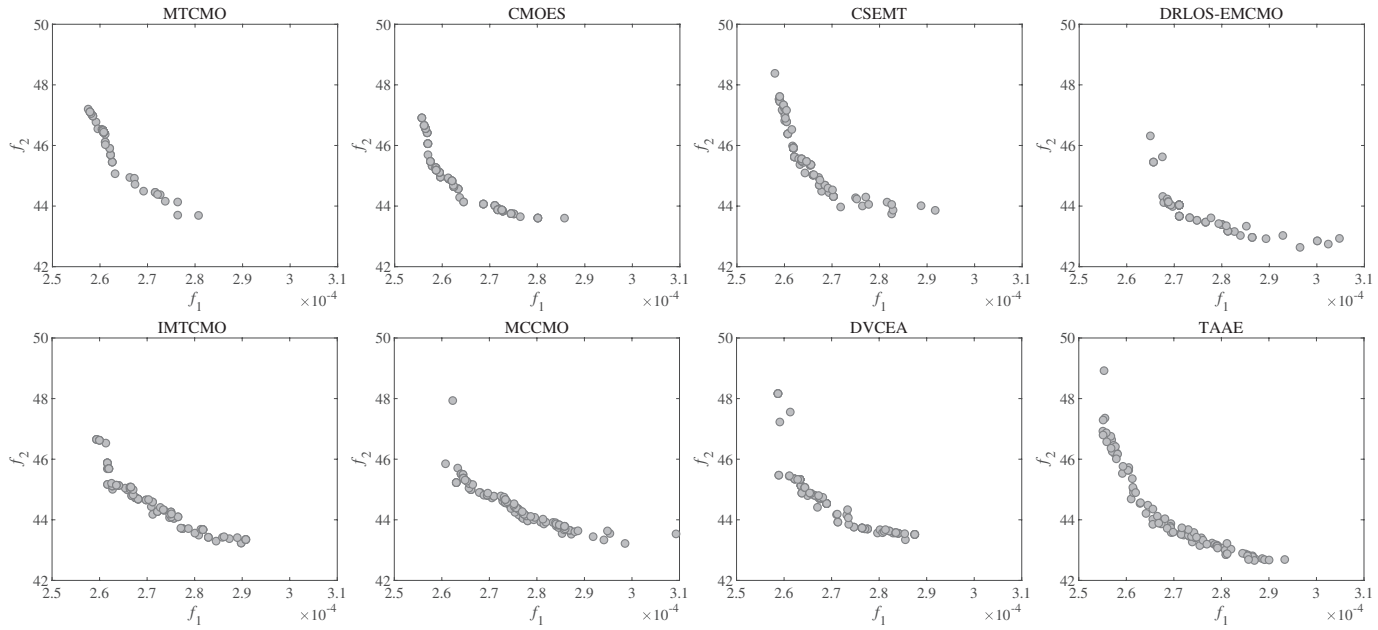


Fig. S47: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS4.

TABLE S19: STATISTICAL RESULTS OF HV OBTAINED BY THE FIVE TAAE VARIANTS WITH DIFFERENT F_{DE}.

Problem	$F_{DE} = 0.1$	$F_{DE} = 0.5$	$F_{DE} = 0.7$	$F_{DE} = 0.9$	$F_{DE} = 0.3$
WS1	1.0419e-1 (9.97e-4) $\approx$	1.0526e-1 (1.22e-3) $\approx$	1.0424e-1 (1.14e-3) $\approx$	1.0421e-1 (7.88e-4) $\approx$	1.0497e-1 (1.66e-3)
WS2	8.0944e-2 (8.40e-4) $\approx$	8.0661e-2 (1.22e-3) $\approx$	8.0869e-2 (1.16e-3) $\approx$	8.0341e-2 (1.05e-3) $\approx$	8.1066e-2 (1.04e-3)
WS3	3.7765e-2 (5.08e-4) $\approx$	3.7698e-2 (5.71e-4) $\approx$	3.7734e-2 (4.28e-4) $\approx$	3.7256e-2 (6.39e-4) $-$	3.7994e-2 (3.04e-4)
WS4	3.3268e-2 (6.37e-4) $\approx$	3.3468e-2 (4.99e-4) $\approx$	3.3449e-2 (4.02e-4) $\approx$	3.3273e-2 (4.94e-4) $\approx$	3.3284e-2 (4.72e-4)
WS5	3.4924e-2 (4.32e-4) $\approx$	3.4854e-2 (5.15e-4) $\approx$	3.5180e-2 (3.45e-4) $\approx$	3.4978e-2 (5.16e-4) $\approx$	3.4989e-2 (3.86e-4)
WS6	2.9403e-2 (5.67e-4) $\approx$	2.9462e-2 (4.82e-4) $\approx$	2.9407e-2 (2.91e-4) $\approx$	2.9236e-2 (4.57e-4) $\approx$	2.9362e-2 (5.17e-4)
WS7	1.6108e-2 (4.17e-4) $\approx$	1.6081e-2 (3.01e-4) $\approx$	1.5990e-2 (4.38e-4) $\approx$	1.6110e-2 (2.93e-4) $\approx$	1.6161e-2 (3.47e-4)
WS8	4.9816e-2 (7.60e-4) $\approx$	4.9732e-2 (7.40e-4) $\approx$	5.0015e-2 (4.14e-4) $\approx$	4.9495e-2 (6.13e-4) $\approx$	4.9669e-2 (5.76e-4)
$+/-/\approx$	0/0/8	0/0/8	0/0/8	0/1/7	

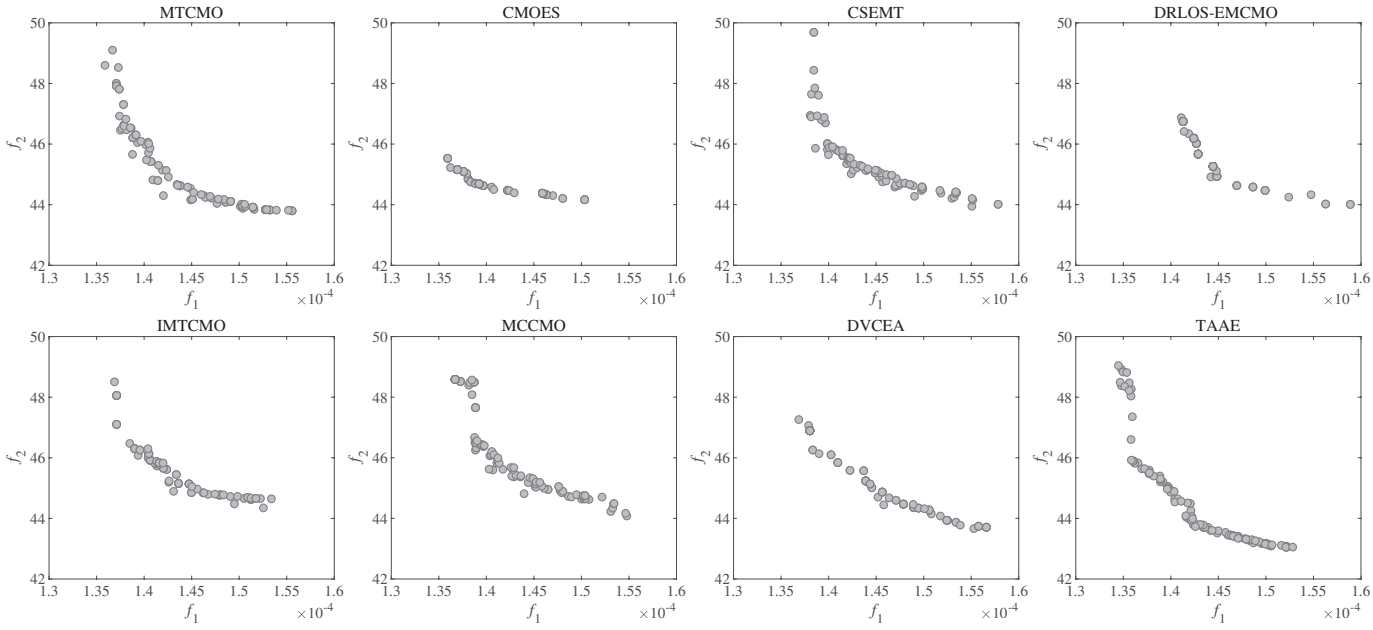


Fig. S48: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS5.

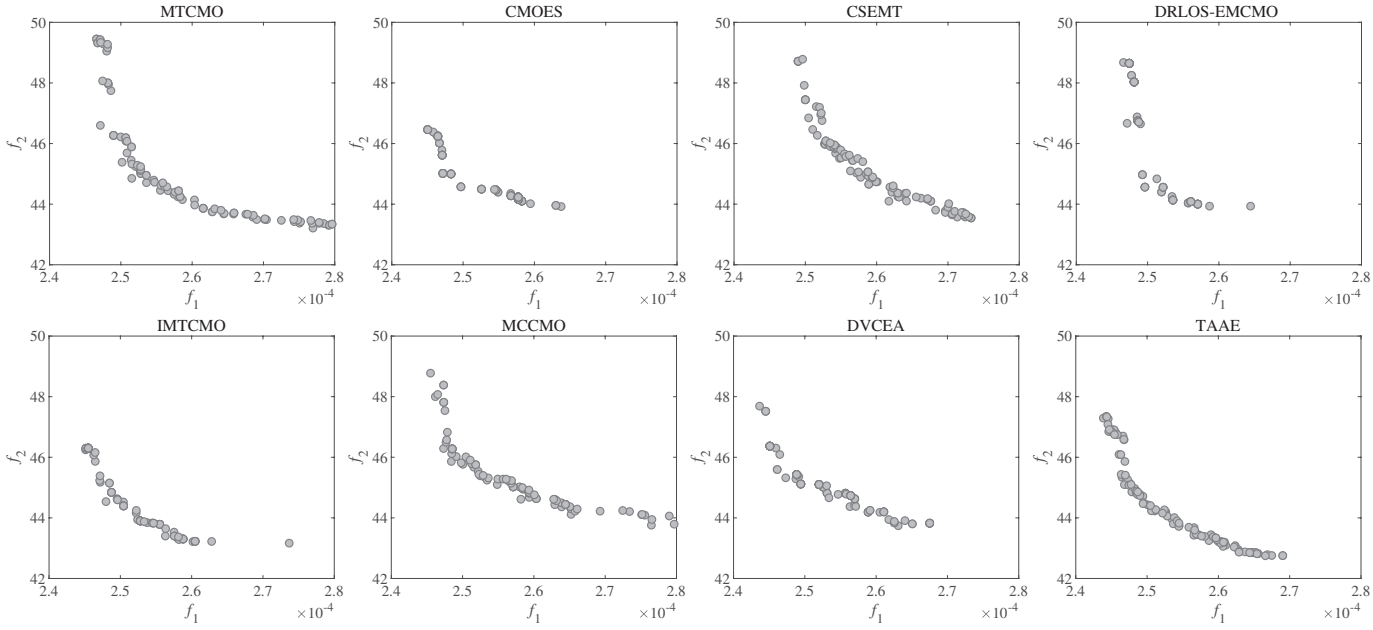


Fig. S49: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS6.

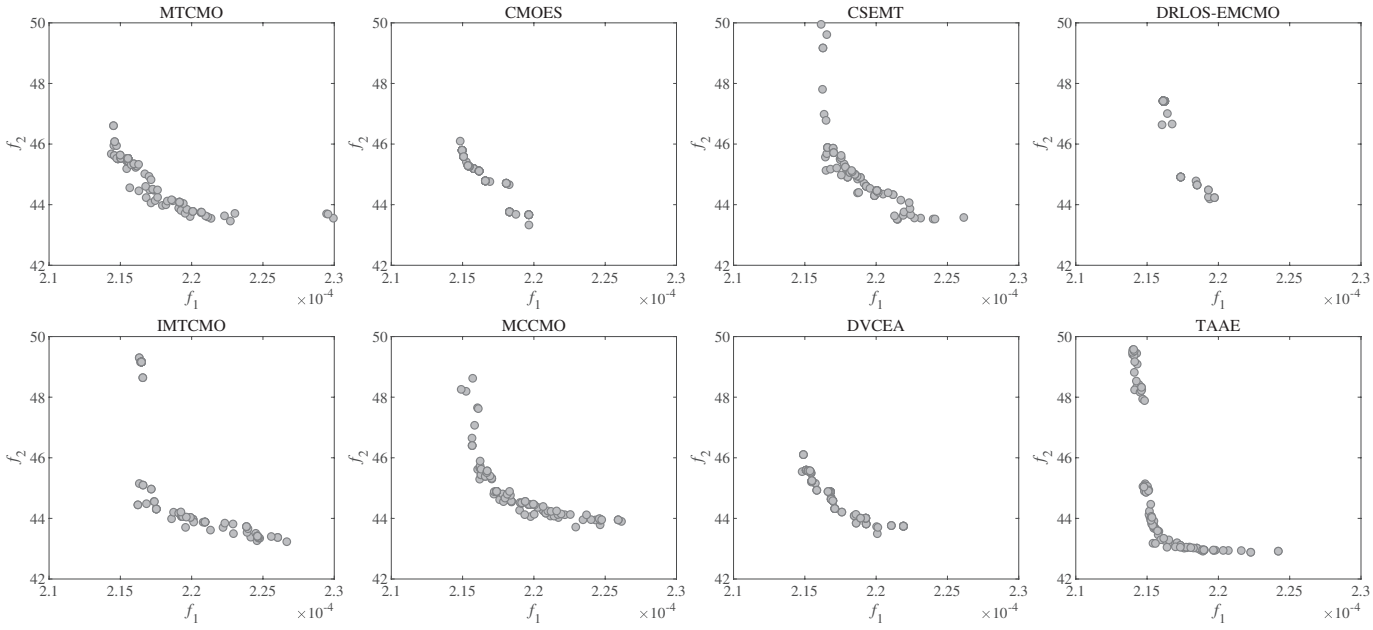


Fig. S50: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS7.

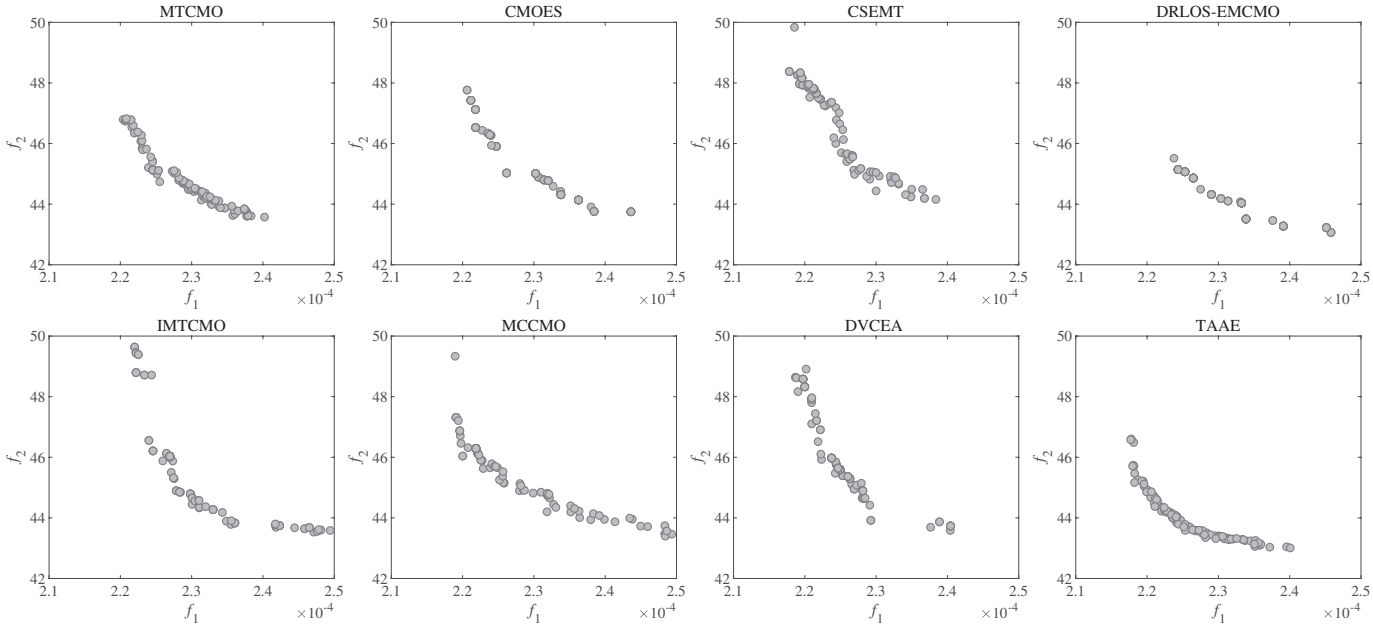


Fig. S51: Final populations obtained by MTCMO, CMOES, CSEMT, DRLOS-EMCMO, IMTCMO, MCCMO, DVCEA, and TAAE on the C6 test suite under WS8.

TABLE S20: STATISTICAL RESULTS OF HV OBTAINED BY THE THREE TAAE VARIANTS WITH DIFFERENT FINE-TUNING EPOCHS.

Problem	$Epoch = 5$	$Epoch = 15$	$Epoch = 10$
WS1	1.0447e-1 (1.14e-3) $\approx$	1.0431e-1 (1.42e-3) $\approx$	1.0497e-1 (1.66e-3)
WS2	8.0611e-2 (1.24e-3) $\approx$	8.0894e-2 (1.16e-3) $\approx$	8.1066e-2 (1.04e-3)
WS3	3.7505e-2 (5.42e-4) $-$	3.7996e-2 (4.41e-4) $\approx$	3.7994e-2 (3.04e-4)
WS4	3.3281e-2 (6.33e-4) $\approx$	3.3276e-2 (5.96e-4) $\approx$	3.3284e-2 (4.72e-4)
WS5	3.4839e-2 (6.18e-4) $\approx$	3.5138e-2 (4.06e-4) $\approx$	3.4989e-2 (3.86e-4)
WS6	2.9205e-2 (4.19e-4) $\approx$	2.9464e-2 (4.02e-4) $\approx$	2.9362e-2 (5.17e-4)
WS7	1.6156e-2 (2.74e-4) $\approx$	1.6138e-2 (2.59e-4) $\approx$	1.6161e-2 (3.47e-4)
WS8	4.9882e-2 (3.93e-4) $\approx$	4.9676e-2 (6.75e-4) $\approx$	4.9669e-2 (5.76e-4)
$+/-/\approx$	0/1/7	0/0/8	

TABLE S21: STATISTICAL RESULTS OF HV OBTAINED BY THE THREE TAAE VARIANTS WITH DIFFERENT LATENT SPACE DIMENSIONS.

Problem	$D_L = 32$	$D_L = 70$	$D_L = 64$
WS1	1.0222e-1 (1.52e-3) $-$	1.0421e-1 (1.82e-3) $\approx$	1.0497e-1 (1.66e-3)
WS2	7.8856e-2 (1.46e-3) $-$	8.0292e-2 (1.53e-3) $\approx$	8.1066e-2 (1.04e-3)
WS3	3.6695e-2 (5.23e-4) $-$	3.7091e-2 (5.76e-4) $-$	3.7994e-2 (3.04e-4)
WS4	3.2836e-2 (3.61e-4) $-$	3.2787e-2 (7.17e-4) $-$	3.3284e-2 (4.72e-4)
WS5	3.4142e-2 (4.06e-4) $-$	3.3752e-2 (8.49e-4) $-$	3.4989e-2 (3.86e-4)
WS6	2.8563e-2 (4.24e-4) $-$	2.8613e-2 (6.21e-4) $-$	2.9362e-2 (5.17e-4)
WS7	1.5712e-2 (3.28e-4) $-$	1.5750e-2 (4.05e-4) $-$	1.6161e-2 (3.47e-4)
WS8	4.8276e-2 (5.14e-4) $-$	4.9048e-2 (7.39e-4) $-$	4.9669e-2 (5.76e-4)
$+/-/\approx$	0/8/0	0/6/2	