Learning-Based Temporal Sequence of Constrained Handling Selection for Constrained Multi-objective Evolutionary Optimization

Chaoda Peng, Siyuan Yan, Cankun Zhong, Qiong Huang, Chunguo Wu, Han Huang, *Senior Member, IEEE*

***Abstract*—Constraint-handling techniques and genetic operators are two crucial components in constrained multi-objective evolutionary algorithms (CMOEAs). Recent research in most of CMOEAs has primarily focused on adaptive designs of these components to address various constrained multi-objective optimization problems (CMOPs). However, the evolutionary process of solving a CMOP can involve various characteristics, such as continuity, discreteness, degeneracy, or some combination thereof, necessitating the tailored selection of constraint-handling techniques and genetic operators across different generations. This study conceptualizes these selections as a temporal sequence of constrained handling selection, where the time means the generation number. We argue that discovering the systematic patterns within the sequence based on the historical data of applying different selections significantly improves the performance of CMOEAs in finding Pareto optimal solutions. Based on this conceptualization, we propose a CMOEA with a deep reinforcement learning model for solving CMOPs. Specifically, the deep reinforcement learning model dynamically refines the selection of constraint-handling techniques and genetic operators for upcoming generations by learning from the performance of previous selections, thereby enhancing the predictive accuracy for subsequent selections. Experiments are conducted to validate the performance of the proposed algorithm against nine CMOEAs on thirty-seven benchmark problems and an unmanned aerial vehicle path planning problem. Experimental results show that the proposed algorithm substantially outperforms the compared algorithms regarding the obtained Pareto optimal solutions. Additionally, the results verify that discovering the systematic patterns within the sequence for CMOEAs has a positive impact on solving CMOPs in terms of objective optimization and constraint satisfaction.**

***Index Terms*—Constrained multi-objective optimization, evolutionary algorithm, temporal sequence of constrained handling selection, deep reinforcement learning**

I. INTRODUCTION

There are a variety of real-world applications that can be attributed to constrained multi-objective optimization problems (CMOPs), such as Internet of Things applications [1], autonomous vehicle path planning problems [2], analog circuit sizing [3], and short-term crude oil scheduling problems [4]. Without loss of generality, a CMOP can be defined as follows:

$$\begin{aligned} &\min F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_m(\mathbf{x}))^T \\ &s.t. \begin{cases} g_i(\mathbf{x}) \leq 0, & i = 1, 2, ..., q \\ h_i(\mathbf{x}) = 0, & i = q+1, q+2, ..., p \\ \mathbf{x} \in \mathbb{D}^c \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{x}$ is a c-dimensional decision vector, $F(\mathbf{x})$ is an m-dimensional objective vector, $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are the inequality and equality constraints, respectively, q is the number of inequality constraints, and p is the total number of constraints. The constraint violation of $\mathbf{x}$ at the i-th constraint is calculated as:

$$\eth_i(\mathbf{x}) = \begin{cases} \max(0, g_i(\mathbf{x})) & 1 \leq i \leq q \\ \max(0, |h_i(\mathbf{x})| - \delta) & q+1 \leq i \leq p \end{cases} \tag{2}$$

where δ is a very small positive value to relax the equality constraints into inequality ones. It is generally set to 0.0001 [5]. The overall constraint violation of $\mathbf{x}$ can be calculated as:

$$G(\mathbf{x}) = \sum_{i=1}^{p} \eth_i(\mathbf{x}) \tag{3}$$

A solution $\mathbf{x}$ is feasible if $G(\mathbf{x})$ is equal to zero; otherwise, the solution is infeasible.

In recent designs of constrained multi-objective evolutionary algorithms (CMOEAs), emphasis has been placed on effectively balancing convergence, diversity, and feasibility [5]–[16]. This balance is crucial because it directly affects the ability of CMOEAs to explore and exploit the feasible regions of the search space while addressing CMOPs. Convergence focuses on guiding solutions towards the Pareto front, which improves their alignment with the objectives. Diversity facilitates a comprehensive exploration of the Pareto front, enabling the generation of a diverse solution set to cater to different preferences. Meanwhile, feasibility is critical because

This work was supported in part by the National Natural Science Foundation of China (62202177, 62276103), in part by the Innovation Team Project of General Colleges and Universities in Guangdong Province (2023KCXTD002), in part by the Guangzhou Basic Research Program (2025A04J5508), in part by the Research and Development Project on Key Technologies for Intelligent Sensing and Analysis of Urban Events Based on Low-Altitude Drones (2024BQ010011), in part by the 2023 Special Program for Audit Theory Research, Guangdong Provincial Philosophy and Social Science Planning (GD23SJZ09), in part by the Fundamental Research Funds for the Central Universities, JLU (93K172024K24), and in part by the Open Research Topics of the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, 2024 (93K172024K26). (Corresponding author: Han Huang)

Chaoda Peng, Siyuan Yan, Cankun Zhong and Qiong Huang are with the College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, China (e-mail: ChaodaPeng@scau.edu.cn; siyuanyan@stu.scau.edu.cn; ck.zhong@scau.edu.cn; qhuang@scau.edu.cn).

Chunguo Wu is with the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, College of Computer Science and Technology, Jilin University, Changchun 130012, China (wucg@jlu.edu.cn).

Han Huang is with the School of Software Engineering, South China University of Technology, Guangzhou 510006, China, and also with the Key Laboratory of Big Data and Intelligent Robot (SCUT), MOE of China, Guangzhou 510006, China (e-mail: hhan@scut.edu.cn).

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2025.3584207

it guarantees that all solutions satisfy the constraints. Li *et al.* [12] proposed a two-archive evolutionary algorithm for solving CMOPs. One archive focuses on feasibility and convergence, while the other aims at promoting diversity. These archives are utilized collaboratively to maintain the balance. Additionally, a restricted mating selection mechanism is designed to enhance the efficacy of the genetic operators. Similarly, Ming *et al.* [16] proposed a dual-population CMOEA with self-adaptive penalties to achieve this balance. Both constraint-handling techniques (CHTs) and genetic operators play essential roles in the design of CMOEAs, with their appropriate selection being vital to maintaining the balance.

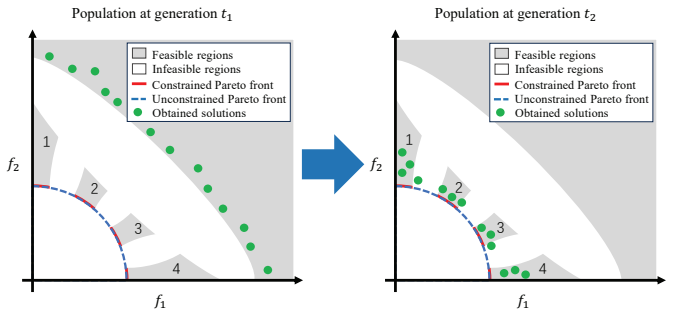


Fig. 1. Illustration of various challenges at two different evolutionary generations t_1 and t_2 $(t_1 < t_2)$.

The selections of CHTs and genetic operators are very challenging during the evolutionary process. The constraints of a CMOP restrict the feasible region in the objective space. This restriction can result in the Pareto front showing various characteristics, such as continuous, discrete, mixed, and degenerate [17], leading to difficulties in continuously adjusting the selections. An example to illustrate this idea is shown in Fig. 1. Suppose a CMOEA encounters an infeasible region at generation t_1, it is imperative to employ a CHT designed to prioritize objective minimization. Note that the constrained Pareto front is distributed on several disjoint feasible regions (i.e., the grey regions labeled with numbers 1–4), utilizing a genetic operator with exploration capabilities is beneficial for the CMOEA to find optimal solutions [17]. By the time the generation number reaches t_2, the CMOEA employs a CHT that prioritizes constraint satisfaction as the population is near the constrained Pareto front. Additionally, the CMOEA can use a genetic operator with strong exploitation capabilities to speed up convergence to the constrained Pareto front. The appropriate selection of CHTs and genetic operators at these two generations can improve the ability of the CMOEA to obtain Pareto optimal solutions.

Recent research on CMOEAs can be grouped into four categories. The first category consists of CMOEAs based on multiple populations [5], [6], [12]–[15], [18]. These algorithms employ multiple populations, each utilizing a unique CHT. The selected CHTs have different focuses in balancing convergence, diversity, and feasibility, enabling cooperation among the various populations. The second category includes CMOEAs based on multiple stages [8], [19]–[23], each of which handles constraints using different CHTs at different stages of the evolutionary process. Generally, these algorithms ignore constraint violations in the early stages and gradually consider feasibility in the latter stages. The third category comprises CMOEAs based on penalty functions [9], [16], [24]. These algorithms incorporate penalty terms into the objective functions, thereby transforming the CMOP into an unconstrained one. The last category consists of CMOEAs based on learning strategies [7], [10], [25]–[27]. These algorithms focus on adaptively selecting strategies for solving CMOPs by utilizing population information, such as the feasibility ratio in the current population, changes in constraint violations, and the relationship between constrained and unconstrained Pareto fronts.

CMOEAs within the previously described four categories do not prioritize the selection of CHTs and genetic operators in each generation. The selections throughout the evolutionary process can be conceptualized as a temporal sequence, where the time refers to the generation number. By leveraging the historical data of applying CHTs and genetic operators, discovering the structured patterns within this sequence is advantageous for a CMOEA in finding Pareto optimal solutions, as the discovery enables the algorithm to identify which selections were the most effective in similar past environments. The primary motivation of this paper is to utilize the historical performance of previous selections to predict the current selection when solving a CMOP with evolutionary algorithms. This facilitation is crucial because the effectiveness of a CMOEA is closely tied to its ability to maintain the balance.

This paper focuses on the temporal sequence of constrained handling selection (see **Definition 1**) for solving CMOPs. The main contributions of this paper are summarized as follows:

- Different from current adaptive CMOEAs, the selections of CHTs and genetic operators throughout the evolutionary process are modeled as a temporal sequence of constrained handling selection in this study. By conceptualizing the selections as such, the proposed algorithm can dynamically adapt to the diverse requirements of balancing convergence, diversity, and feasibility encountered during the evolutionary process, thereby enhancing its performance in finding Pareto optimal solutions. It is noteworthy that, from an essential perspective of adaptively designing CHTs and genetic operators for solving CMOPs, other CMOEAs can be considered special instances within the framework of the proposed model.
- We argue that discovering the systematic patterns in the temporal sequence of constrained handling selection is beneficial for CMOEAs in finding Pareto optimal solutions. To facilitate this discovery, a deep reinforcement learning model is designed in this study. This model leverages historical data of previous selections to predict the most effective selection in each generation. By continuously learning from past experiences, the model dynamically refines its strategies, improving predictive accuracy and decision-making effectiveness. Experimental results substantiate our argument, demonstrating that discovering the systematic patterns within the temporal sequence of constrained handling selection positively impacts the resolution of CMOPs.

The remainder of this paper is organized as follows. Section II reviews the recent research about constrained multi-objective optimization. Section III introduces the temporal sequence of constrained handling selection method. Section IV provides a detailed introduction of the proposed CMOEA based on the temporal sequence of constraint-handling selection methods. Section V presents experimental results. Section VI provides the conclusion and outlines future research directions.

II. RELATED WORK

In this section, we present recent advancements in constrained multi-objective optimization, organized according to the four categories described in Section I.

CMOEAs based on multiple populations utilize the cooperation of multiple populations. Qiao *et al*. [6] proposed a multitasking constrained multi-objective optimization framework for solving CMOPs. This algorithm employs a dynamic auxiliary population to facilitate the evolution of the main population through knowledge transfer. Ming *et al*. [15] presented a competitive and cooperative swarm optimizer for solving CMOPs. The competitive swarm optimizer aims to approximate the Pareto front, while the cooperative swarm optimizer focuses on escaping local optima. Liu *et al*. [14] devised a novel bidirectional coevolution CMOEA with one main population and one archive population. The main population is designed to prioritize constraint satisfaction, and the archive population is designed to focus on objective minimization. Although these CMOEAs employ multiple cooperative CHTs across the populations, the selection of these CHTs is predetermined. Moreover, the selection remains unchanged throughout the evolutionary process, thereby limiting the performance of CMOEAs in finding Pareto optimal solutions.

CMOEAs based on multiple stages segment the evolutionary process into distinct stages, each employing a distinct CHT. Fan *et al*. [19] proposed a CMOEA with a push stage and a pull stage. The algorithm disregards constraints at the push stage, whereas it uses an improved ε method to guide the population back to feasible regions at the pull stage. To switch the push stage to the pull stage, a switching condition is designed based on the change of both the ideal and the nadir points. Zhang *et al*. [21] proposed a two-stage multi-objective evolutionary process to solve CMOPs, designing a parameter-less CHT to decompose the entire population into three subsets. The design of the CHT aims to balance convergence and constraint satisfaction across different subsets at various stages. Most of these CMOEAs adopt a specific CHT at a stage. Therefore, they have difficulty in adapting to the generation-varying requirements for the balance at this stage. Besides, the use of the CHTs during different stages is manually determined, which is labor-intensive and problem-dependent.

CMOEAs based on penalty function use a penalty term to penalize infeasible solutions. Ma *et al*. [24] proposed a new fitness function based on the constraint domination principle (CDP) to solve CMOPs. The algorithm uses the weighted sum of two rankings based on CDP and Pareto domination to evaluate the superiority of the solutions, where the weight is related to the proportion of feasible solutions in the current population. Maldonado *et al*. [28] proposed a dynamic penalty function within MOEA/D for CMOPs. The penalty term is dynamically adjusted along the evolutionary process, so as to utilize the genetic information from the interaction between the feasible and infeasible solutions.

CMOEAs based on learning strategy utilize the information generated during the evolutionary process for adaptive decision-making. Liang *et al*. [7] designed a CMOEA based on the utilization of the relationship between the unconstrained and constrained Pareto fronts. The evolutionary process is divided into the learning and evolving stages. The purpose of the learning stage is to measure the relationship between the two Pareto fronts. Subsequently, the information obtained at the learning stage is used to choose a better evolving strategy for the evolving stages. Peng *et al*. [26] proposed a two-stage framework for locating the reference point in the decomposition-based CMOEAs. By learning the approximate locations of the unconstrained and constrained Pareto fronts at the first stage, a local estimation mechanism is designed to estimate the best fit location of the reference point for a CMOEA at the second stage. Most of these CMOEAs operate under the assumption that different constrained handling strategies exhibit distinct performance levels when solving specific CMOPs. To maintain overall performance across a variety of CMOPs, these algorithms incorporate adaptive mechanisms that accommodate the diverse characteristics of CMOPs, thereby enhancing their robustness in addressing the CMOPs effectively.

Recently, some studies use evolutionary algorithms with reinforcement learning to solve CMOPs [10], [27]. To alleviate the burden of selecting genetic operators, Zou *et al*. [10] introduced a process knowledge-guided strategy that employs deep reinforcement learning. The strategy automatically recommends a suitable genetic operator to the CMOEA (PKAEO). Ming *et al*. [27] presented an adaptive auxiliary task selection for multitasking-assisted constrained multi-objective optimization with the utilization of the Q-Learning (CMOQLMT) and deep Q-Learning models (CMODQLMT). The reinforcement learning models intelligently suggest the optimal knowledge transfer auxiliary task. Compared to the CMOEAs that utilize reinforcement learning for selecting CHTs or genetic operators to address CMOPs, this study conceptualizes the selections as a temporal sequence of constrained handling selection from an essential perspective (see **Definition 1**). This conceptualization is significant for enhancing the performance of CMOEAs in identifying Pareto optimal solutions. Additionally, this study enables the comprehensive selections of both CHTs and genetic operators for the proposed CMOEA, improving the adaptability and overall effectiveness of the CMOEA to dynamically respond to the generation-varying landscapes of a CMOP.

III. TEMPORAL SEQUENCE OF CONSTRAINED HANDLING SELECTION

This section presents the foundation related to the temporal sequence of constrained handling selection. Moreover,

a method that leverages this sequence is introduced in detail, aiming to enhance the adaptability and effectiveness of CMOEAs in finding Pareto optimal solutions. For a quick reference to the parameters defined in this study, we provide a summary in Tables S-I and S-II in the supplementary document.

A. Foundation about the temporal sequence of constrained handling selection

The concept of the temporal sequence of constrained handling selection refers to a sequence of the constraint-handling strategies throughout the evolutionary process. For a CMOEA and CMOP, let $\mathbb{P}_1$ represent the initial population with size N, $T_{\max}$ denote the maximum number of generations, and $\mathbb{A}$ denote a set of constraint-handling strategies with size I. The definition of the temporal sequence of constrained handling selection is given as follows:

Definition 1. (**Temporal Sequence of Constrained Handling Selection**). A series $\Upsilon := [a_1, a_2, \ldots, a_{T_{\max}}]$ is called a temporal sequence of constrained handling selection, where each a_t represents a CHT with a genetic operator in the t-th generation, and $a_t \in \mathbb{A}$. The sequence is recursively defined as follows:

$$a_t := \mho\left(a_1, a_2, \ldots, a_{t-1}\right) \tag{4}$$

where $\mho$ is a predictive model that is used to output the next selection a_t in $\mathbb{A}$ based on the historical sequence of the constrained handling strategies, and $t = 2, 3, \ldots, T_{\max}$. The initial constrained handling strategy a_1 is randomly selected from $\mathbb{A}$ following a uniform distribution.

Based on **Definition 1**, current CMOEAs with designing adaptive CHTs and operators can be considered as special instances within the proposed model of the temporal sequence of constrained handling selection. For example, CMOEAs based on multiple stages adopt multiple CHTs at different stages of the evolutionary process. The adoption of the CHTs throughout the evolutionary process forms a temporal sequence of constrained handling selection, where each stage corresponds to a specific segment of the temporal sequence in our model. Our model generalizes these CMOEAs by providing a unified structure that captures the dynamic and sequential nature of constrained handling selection throughout the evolutionary process.

B. Temporal sequence of constrained handling selection based on deep reinforcement learning

This subsection presents the framework of the temporal sequence of constrained handling selection based on deep reinforcement learning and introduces how to utilize the deep Q-network to seek the systematic patterns within the sequence. As shown in Fig. 2, the primary distinction between the proposed algorithm and other CMOEAs that utilize deep reinforcement learning is that our algorithm facilitates comprehensive selections of both CHTs and genetic operators in an online manner. The selection necessitates modifications in key aspects of the deep reinforcement learning design, including state representation, action space, and reward scheme. These tailor-designed components enable our algorithm to adapt more effectively to the dynamic requirements of solving CMOPs, resulting in better performance in finding Pareto optimal solutions compared with the existing deep reinforcement learning-based CMOEAs.

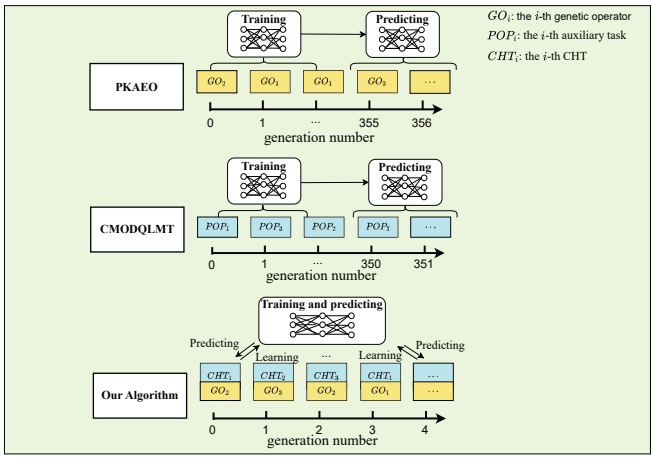


Fig. 2. Comparison between the proposed algorithm and two recent CMOEAs based on deep reinforcement learning for adaptively selecting constrained handling strategies.

Considering that a CHT and genetic operator in each generation are selected from $\mathbb{A}$ according to **Definition 1**, the total number of possible selection paths scales exponentially with the number of generations. Therefore, identifying the systematic patterns in the sequence is a computationally demanding task. In this context, reinforcement learning plays a crucial role in mitigating this computational challenge. It enables the agent to dynamically refine its strategies based on feedback from the evolutionary process, thereby enhancing decision-making within CMOEAs by focusing on empirically effective strategies. Consequently, reinforcement learning streamlines the search process and provides a feasible method to reduce the complexity involved in identifying the systematic patterns within the sequence for CMOEAs.

In this paper, the deep Q-network is utilized to seek the structured patterns in the temporal sequence of constrained handling selection, as shown in Fig. 3. The deep Q-network characterizes the selection of CHTs and genetic operators as action, with specific population information serving as the state. Besides, it measures the improvement in objective optimization and constraint satisfaction between two consecutive populations to determine the reward. During the evolutionary process, the proposed algorithm adjusts the CHT and genetic operator based on the reward in each generation, generating data samples for training the deep Q-network. As a result, the deep Q-network can become more accurate for predicting selections.

In **Algorithm 1**, the state, denoted as $\boldsymbol{s}_t$, is obtained based on population $\mathbb{P}_t$ (Lines 1-2). To balance exploration and exploitation, an action, denoted as a_t, is randomly selected with a small probability using an Epsilon-greedy strategy [29],

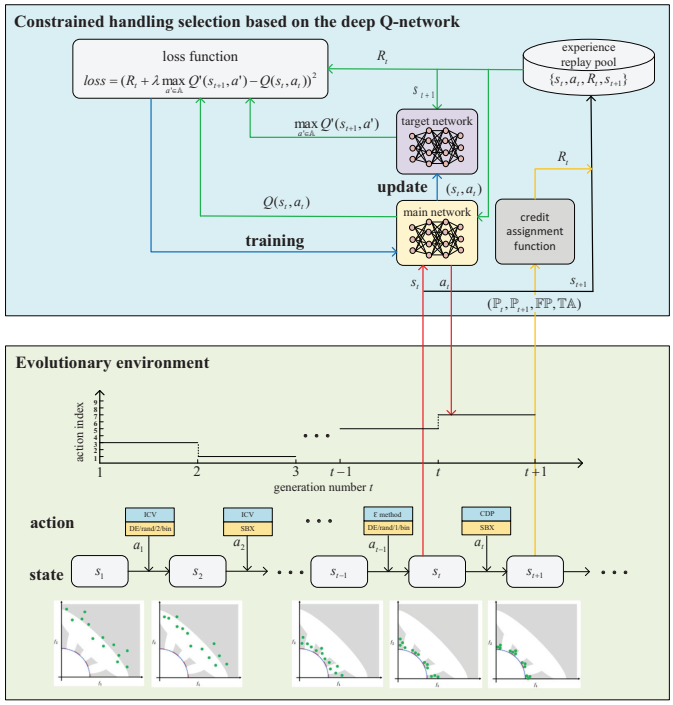


Fig. 3. Framework of the temporal sequence of constrained handling selection based on the deep Q-network.

Algorithm 1: Temporal sequence of constrained handling selection based on the deep Q-network

Input: The population $\mathbb{P}_t$ at generation t, the main network Q, the set of actions $\mathbb{A}$, the generation number t

Output: The selected action a_t, the state $\boldsymbol{s}_t$

/*Equation (12) is derived from Equations (5)-(11) */
Obtain $\boldsymbol{s}_t$ of $\mathbb{P}_t$ according to Equation (12);
/*ξ is a predefined value between 0 and 1 */
if $rand < \xi$ *or* $t < b$ **then**
  Randomly select an index i from 1 to $\|\mathbb{A}\|$ with a uniform distribution;
  $a_t \leftarrow$ the i-th element in $\mathbb{A}$;
else
  /*pick up the action with the maximum Q-value */
  Obtain $\|\mathbb{A}\|$ Q-values by inputting $\boldsymbol{s}_t$ to Q;
  $a_t \leftarrow \arg\max_{a' \in \mathbb{A}} Q(\boldsymbol{s_t}, a')$;
end
return $\boldsymbol{s}_t$, a_t

particularly when the number of training samples is less than the batch size, denoted as b. This ensures that the training process always begins with at least a full batch of data (Lines 3-6). If there are enough samples, $\boldsymbol{s}_t$ serves as the input to the main network, denoted as Q, within the deep Q-network. This input generates a set of distinct Q-values, each associated with a specific action. Subsequently, the action a_t with the maximum Q-value is selected as the optimal action (Lines 8-10).

We discuss the key components, i.e., the state, action, reward, training, and updating of the deep Q-network in the following subsections.

1) Design of the state: In this paper, feasibility, convergence, and diversity are utilized to characterize the state in each generation.

Two features are used to characterize the feasibility of the population. One is the ratio of the feasible solutions in population $\mathbb{P}_t$, as shown in Equation (5).

$$\gamma_t = \frac{\vartheta_t}{||\mathbb{P}_t||} \tag{5}$$

where ϑ_t denotes the number of feasible solutions in $\mathbb{P}_t$. Another is the ratio of the feasible non-dominated solutions over the non-dominated solutions in $\mathbb{P}_t$, as shown in Equation (6).

$$\tau_t = \frac{\sigma_t}{\omega_t} \tag{6}$$

where σ_t and ω_t denote the number of feasible non-dominated solutions and non-dominated solutions in $\mathbb{P}_t$, respectively.

Two features are adopted to describe the population convergence. The ideal point, denoted as $\boldsymbol{z}^*$, can be used to measure how far a population is close to the regions with the minimum objective values. It is given as follows:

$$\boldsymbol{z}^* = (z_1, z_2, \ldots, z_m) \tag{7}$$

where $z_i = \min\left(f_i(\mathbf{x}) | \mathbf{x} \in \bar{\mathbb{P}}\right)$, $\bar{\mathbb{P}}$ is the set of all individuals found so far. Besides, the center point, denoted as $\boldsymbol{\nu_t}$, of $\mathbb{P}_t$ can be used to reflect the position where a population reaches along the evolutionary process. It is given as follows:

$$\boldsymbol{\nu_t} = \left(\nu_t^1, \nu_t^2, \ldots, \nu_t^m\right) \tag{8}$$

where $\nu_t^i = \frac{\sum_{\mathbf{x} \in \mathbb{P}_t} f_i(\mathbf{x})}{||\mathbb{P}_t||}$, and $i = 1, 2, \ldots, m$.

Population diversity plays a pivotal role in characterizing the state, as different degrees of population diversity necessitate varying constraint-handling strategies [30]. Two metrics are adopted to quantify population diversity, i.e., the average distance between $\mathbb{P}_t$ and $\boldsymbol{\nu_t}$, denoted as κ_t, and standard deviation of $\mathbb{P}_t$, denoted as ρ_t. They are given in Equations (9) and (10), respectively.

$$\kappa_t = \frac{\sum_{\mathbf{x} \in \mathbb{P}_t} d(F(\mathbf{x}), \boldsymbol{\nu_t})}{||\mathbb{P}_t||} \tag{9}$$

$$\rho_t = \sqrt{\frac{\sum_{\mathbf{x} \in \mathbb{P}_t} \left(d(F(\mathbf{x}), \boldsymbol{\nu_t}) - \kappa_t\right)^2}{||\mathbb{P}_t||}} \tag{10}$$

where $d(F(\mathbf{x}), \boldsymbol{\nu_t})$ calculates the Euclidean distance between $F(\mathbf{x})$ and $\boldsymbol{\nu_t}$.

A binary control parameter, denoted as $flag$, is employed to manage transitions between two distinct stages of the evolutionary process. The reason is that each stage in the evolutionary process is characterized by distinct requirements

on balancing feasibility, convergence, and diversity. The parameter $flag$ is designed to reflect the update ratio of the population every ten generations, as defined in Equation (11). The algorithm is in the exploration stage, with a focus on expanding the search to explore a diverse range of solutions when $flag$ is equal to 0. Conversely, the population is roughly considered to have reached the unconstrained Pareto front when $flag$ is switched to 1.

$$flag = \begin{cases} 1, & if\ \mu_j < \mu_{\text{thr}} \text{ for all } j \in J \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where μ_j represents the ratio of individuals in the temporary archive, denoted as $\mathbb{TA}$, that are updated by the offspring population in each update scheme (see Lines 24-25 in **Algorithm 2**). Specifically, μ_j is calculated as the number of new individuals that replace older ones in $\mathbb{TA}$ divided by the total number of individuals in $\mathbb{TA}$. The archive $\mathbb{TA}$ stores the best non-dominated individuals identified by the offspring. Furthermore, μ_{thr} is a preset threshold, and J represents a set comprising every ten consecutive generation numbers.

The state $\boldsymbol{s}_t$ for the deep Q-network is formulated as a vector composed of the seven aforementioned features, in which the features are calculated by Equations (5)-(11). The state $\boldsymbol{s}_t$ is represented as follows:

$$\boldsymbol{s}_t = (\gamma_t, \tau_t, \boldsymbol{z}^*, \boldsymbol{\nu_t}, \kappa_t, \rho_t, flag) \tag{12}$$

2) Design of the action: Both CHTs and genetic operators play a significant role in solving CMOPs [31]. As for CHTs, there are a lot of CHTs enabling CMOEAs to well handle different kinds of CMOPs [22], [32]. In general, they can be grouped into three categories based on their sensitivity to the constraint violations. The first group is characterized by its insensitivity to constraint violations, causing the CMOEAs to completely ignore constraint violations. The second group exhibits semi-sensitivity to constraint violations. The ε method is a representative CHT which enables the CMOEAs to choose some individuals with tolerable constraint violations to the next generation. The last group is characterized by its complete sensitivity to constraint violations, as it prioritizes feasible solutions all the time. CDP is a prime example of this category. To enhance the capability of the proposed algorithm in addressing a diverse range of CMOPs with various characteristics, one representative CHT is taken from each of the three groups, i.e., ignoring constraint violations (ICV) [19], the ε method [6], and CDP [33].

As for genetic operators, differential evolution (DE) operators and simulated binary crossover (SBX) [34] are the most commonly used to solve CMOPs recently due to their strong search ability [6], [16]. Several types of DE operators have shown excellent ability to solve optimization problems. DE/rand/1/bin and DE/rand/2/bin are two of them that enable fine-tuning of the search process by adjusting parameters to improve the population diversity [10]. Consequently, both are integrated as key genetic operators in the proposed algorithm to leverage their distinct advantages. SBX is another genetic operator that is chosen in this study, as it has a different working paradigm with DE operators.

The action set, denoted as $\mathbb{A}$, consists of nine unique actions, each formed by pairing a CHTs with a genetic operators, as shown in Table I. The actions are indexed from 1 to 9, where each index number corresponds to a specific pair of the CHT and genetic operator. The details of the CHTs and genetic operators used in this study are presented in Section I-A and Section I-B in the supplementary materials, respectively.

TABLE I
DESIGN OF THE ACTION IN THIS STUDY.

Action	CHT	Genetic operator
1	ICV	SBX
2	ICV	DE/rand/1/bin
3	ICV	DE/rand/2/bin
4	ε method	SBX
5	ε method	DE/rand/1/bin
6	ε method	DE/rand/2/bin
7	CDP	SBX
8	CDP	DE/rand/1/bin
9	CDP	DE/rand/2/bin

3) Design of the reward: The reward evaluates the value of the action taken, assigning the maximum value to the best action. In this study, we design an adaptive credit assignment function to dynamically assign a reward to an action based on the state in each generation. The reward is designed in two phases to guide the proposed algorithm to explore the search space at the beginning of the evolutionary process and subsequently to focus on finding Pareto optimal solutions. The reward, denoted as R_t, is presented as follows:

$$R_t = \begin{cases} (IGD_t - IGD_{t+1}) \cdot 10^3, & flag = 0 \\ (1 - \lfloor \gamma_{t+1} \rfloor) \cdot lb \frac{\iota_{t+1}}{\iota'} + \lfloor \gamma_{t+1} \rfloor \cdot \eta \frac{\rho_{t+1}}{\rho'}, & \text{otherwise} \end{cases} \tag{13}$$

where IGD_t and IGD_{t+1} represent the inverted generational distance (IGD) [35] of $\mathbb{P}_t$ and $\mathbb{P}_{t+1}$ according to Equation (14), respectively. $\lfloor \gamma_{t+1} \rfloor$ denotes the floor function applied to the feasibility ratio in $\mathbb{P}_{t+1}$. η is the number of the updated individuals in the output archive storing the best N feasible non-dominated solutions, denoted as $\mathbb{FP}$, at generation t. ι_{t+1} and ι' are the sum of the overall constraint violations on all individuals in $\mathbb{P}_{t+1}$ and $\mathbb{TA}$, respectively. ρ_{t+1} and ρ' are the standard deviations of $\mathbb{P}_{t+1}$ and $\mathbb{TA}$ based on Equation (10), respectively. lb and ub are the lower and upper bound for the reward, respectively. lb is less than 0 and ub is greater than 0. It is worth noting that the reward is limited to the domain $[lb, ub]$. The reward is capped at the lower bound if it falls below lb or at the upper bound if it exceeds ub. The reason is that the reward is taken as one of the features for training the deep Q-network, and its value that is outside of the usual range can cause large gradients to back-propagation, resulting in permanently shutting of activation functions due

to the vanishing gradients [36].

$$IGD_t = \frac{\sum_{\mathbf{x}\in\mathbb{TA}} md(F(\mathbf{x}), \mathbb{P}_t)}{||\mathbb{TA}||} \tag{14}$$

The Equation (14) calculates the Euclidean distance between the individual $\mathbf{x}$ in $\mathbb{TA}$ and its closest individual in $\mathbb{P}_t$ based on their objective values.

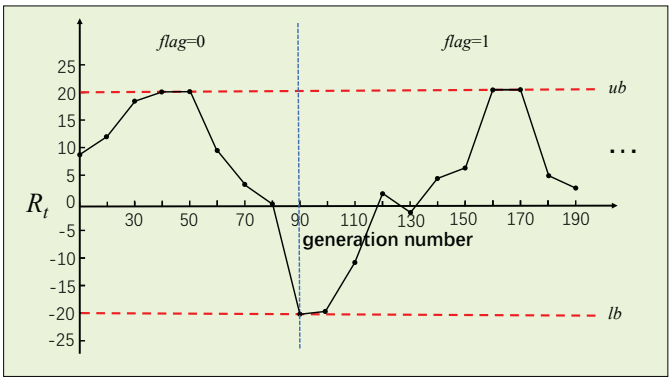


Fig. 4. Example regarding the working paradigm of the reward during the evolutionary process.

To understand the working paradigm of the reward during evolutionary process, an example of how the change of the reward is associated with the evolutionary process is given in Fig. 4. The reward mechanism designed in this study consists of two phases. In the first phase (i.e., when $flag = 0$), the reward is determined by the improvement between two consecutive populations. As the population approaches the unconstrained Pareto front, the rewards gradually decrease because the two consecutive populations become similar. When $flag$ switches to 1, the proposed algorithm enters the second phase, aiming to find Pareto optimal solutions. In this phase, the algorithm engages with two scenarios. In the first scenario where there exists at least an infeasible individual in $\mathbb{P}_{t+1}$, the reward is determined by the sum of constraint violations on all individuals in $\mathbb{P}_{t+1}$ and $\mathbb{TA}$. In the second scenario where all individuals in $\mathbb{P}_{t+1}$ are feasible, the reward is determined by the number of updated individuals in the output archive $\mathbb{FP}$ using the offspring of $\mathbb{P}_t$, diversity of the populations $\mathbb{P}_{t+1}$, and diversity of $\mathbb{TA}$.

4) Training and updating of the deep Q-network: Training and updating the deep Q-network are essential processes designed to continuously enhance its capacity to predict and evaluate action outcomes effectively. The deep Q-network comprises two networks, i.e., the main network, denoted as Q, and target network, denoted as Q'. The main network Q learns and predicts the Q-values for different actions, while Q' serves to stabilize the training process [37]. The initialization of Q involves setting up its architecture with random weights, which are then refined through training. The experience regarding the selection of the CHTs and genetic operators in each generation can be used to train and update the two networks for more accurate predictions. For instance, the next population $\mathbb{P}_{t+1}$ is produced through the action a_t applied to the current population $\mathbb{P}_t$ at generation t. An experience sample, denoted as $\mathbf{e}_t$, is produced during the process. It consists of four components, i.e., $\boldsymbol{s}_t$, a_t, R_t and $\boldsymbol{s}_{t+1}$, where a_t is the selection of the CHT and genetic operator based on Table I, R_t is the reward obtained according to Section III-B3, $\boldsymbol{s}_t$ and $\boldsymbol{s}_{t+1}$ are the states of $\mathbb{P}_t$ and $\mathbb{P}_{t+1}$, respectively. The experience sample is collected into an experience replay pool, denoted as $\mathbb{E}$. where $\mathbb{E}$ is a queue that operates on a first-in-first-out basis in this study. It is worth noting that the head sample is removed when $\mathbb{E}$ exceeds its pre-defined cardinality. The deep Q-network uses the samples from $\mathbb{E}$ to train Q by randomly selecting a batch of samples with size b when there are enough samples, i.e., generation number t is no less than b. A gradient descent method is adopted to train Q for w epochs by using the b selected samples with the loss function by Equation (15).

$$loss = (R_t + \lambda \max_{a'\in\mathbb{A}} Q'(\boldsymbol{s}_{t+1}, a') - Q(\boldsymbol{s}_t, a_t))^2 \tag{15}$$

where λ is a discount factor for the future reward, $\max_{a'\in\mathbb{A}} Q'(\boldsymbol{s}_{t+1}, a')$ is the maximum Q-value of all actions output by Q' at the state $\boldsymbol{s}_{t+1}$, $Q(\boldsymbol{s}_t, a_t)$ is the output of Q at the state $\boldsymbol{s}_t$. The target network Q' undergoes periodic updates, which involve copying the weights from Q to Q' every fixed number of generations, as specified by the update frequency parameter. This process ensures that Q' gradually adapts to the improved estimates of Q while providing a stable baseline for computing the loss during training.

IV. CMOEA BASED ON THE PROPOSED TEMPORAL SEQUENCE

This section introduces the details of the proposed CMOEA based on the temporal sequence of constrained handling selection. Additionally, an analysis of the computational time complexity is provided.

A. Main framework of the proposed CMOEA

The pseudo-code of the proposed CMOEA based on the temporal sequence of constrained handling selection is presented in **Algorithm 2**.

1) Discovering the systematic patterns within the temporal sequence of constrained handling selection: The deep Q-network is facilitated to refine the selection of CHTs and genetic operators in each generation. Moreover, it is trained and updated to be more accurate in predicting the selections by utilizing the data acquired through the interaction with the evolutionary environment.

In each generation (Line 9), the population $\mathbb{P}_t$, main network Q, action set $\mathbb{A}$, and generation number t serve as the input for **Algorithm 1**. The best action, comprising a pair of CHT and genetic operator, denoted as CHT and GO respectively, is selected for its highest expected cumulative reward. Consequently, the deep Q-network leverages the accumulated experience to identify the most effective selection. According to Section III-B4, the network is consistently trained and updated in each generation (Lines 33-34), which enhances the accuracy of the CHTs and genetic operators selected by the proposed algorithm.

Algorithm 2: Main framework of the proposed algorithm

Input: An initial population $\mathbb{P}_1$ with size N, the maximum generation number $T_{\max}$
Output: The set of Pareto optimal solutions $\mathbb{FP}$

1 Initialize the main network Q in the deep Q-network;
2 $Q^{'} \leftarrow Q$; /* Target network */
3 $\mathbb{TA} \leftarrow \mathbb{P}_1$; /* Temporary archive */
4 $\mathbb{FP} \leftarrow \mathbb{P}_1$; /* Output archive */
5 $\mathbb{E} \leftarrow \emptyset$. /* Experience replay pool.*/
6 $t \leftarrow 1$; /* Generation number */
7 **while** $t \leq T_{max}$ **do**
8 /* The action a_t is an index from 1 to 9, each index represents a combination of a genetic operator GO and CHT CHT, as in Table I */
9 $\boldsymbol{s}_t$, $a_t \leftarrow$ **Algorithm 1** $(\mathbb{P}_t, Q, \mathbb{A}, t)$;
10 $rand \leftarrow$ a random value between 0 and 1 with
11 a uniform distribution;
12 **for** $k \leftarrow 1$ *to* N **do**
13 **if** $rand < 0.9$ **then**
14 $\mathbb{P}_{\text{parent}} \leftarrow$ the mating parents selected from the neighbors of the k-th individual via the binary tournament selection;
15 **else**
16 $\mathbb{P}_{\text{parent}} \leftarrow$ the mating parents selected from $\mathbb{P}_t$ and $\mathbb{FP}$ via the binary tournament selection;
17 **end**
18 $op_k \leftarrow$ an offspring generated by $\mathbb{P}_{\text{parent}}$ via GO in a_t according to Table I;
19 $\mathbb{NP} \leftarrow \mathbb{P}_t \cup \{op_k\}$;
20 $\mathbb{P}_{t+1} \leftarrow$ the N individuals selected from $\mathbb{NP}$ based on CHT in a_t according to Table I;
21 /* Update archives $\mathbb{FP}$ and $\mathbb{TA}$ */
22 $\mathbb{NF} \leftarrow \mathbb{FP} \cup \{op_k\}$;
23 $\mathbb{FP} \leftarrow$ the N individuals selected from $\mathbb{NF}$ based on CDP;
24 $\mathbb{NT} \leftarrow \mathbb{TA} \cup \{op_k\}$;
25 $\mathbb{TA} \leftarrow$ the N individuals selected from $\mathbb{NT}$ based on ICV;
26 **end**
27 /* Calculate reward and obtain $\boldsymbol{s}_{t+1}$ */
28 $R_t \leftarrow$ the reward obtained according to Equation (13) and $\mathbb{TA}$.
29 Obtain $\boldsymbol{s}_{t+1}$ of $\mathbb{P}_{t+1}$ according to Equation (12);
30 /* Collect samples into $\mathbb{E}$ */
31 $\mathbf{e}_t \leftarrow \{\mathbf{s}_t, a_t, R_t, \mathbf{s}_{t+1}\}$.
32 $\mathbb{E} \leftarrow \mathbb{E} \cup \{\mathbf{e}_t\}$.
33 /* Train and update the networks with $\mathbb{E}$ */
34 Train Q and update Q' by Section III-B4;
35 $t \leftarrow t + 1$;
36 **end**
37 **return** $\mathbb{FP}$

2) Evolutionary framework based on the temporal sequence of constrained handling selection: The evolutionary algorithm receives the action from the deep Q-network. Subsequently, the genetic operator GO is utilized to generate the offspring population, while CHT is applied to select the subsequent generation within the population.

The binary tournament selection mechanism is utilized to choose the mating parents from the neighbors of each individual in the reproduction scheme. The neighbor relationship is defined by an angle-based decomposition method [38], which is given as follows:

$$\mathbb{B}_k = \{\mathbf{x}_v | \Phi\left(F\left(\mathbf{x}_k\right), F\left(\mathbf{x}_v\right)\right) \leq \theta, \forall \mathbf{x}_v \in \mathbb{P}_t \cup \mathbb{FP}\} \quad (16)$$

where $\mathbb{B}_k$ represents the neighbors of the individual $\mathbf{x}_k$ in $\mathbb{P}_t$, $\Phi\left(F\left(\mathbf{x}_k\right), F\left(\mathbf{x}_v\right)\right)$ is the angle between vectors $F\left(\mathbf{x}_k\right)$ and $F\left(\mathbf{x}_v\right)$, and θ is a preset angle. An offspring op_k is obtained by applying GO in a_t to the k-th individual in $\mathbb{P}_t$ (Lines 10-18). A selection scheme is performed on $\mathbb{P}_t$ with each newly generated individual op_k based on CHT in a_t (Line 20). The selection is influenced by the used CHT. For instance, a CMOEA selects individuals only based on objective values if it uses ICV, i.e., completely ignoring the constraint violations. The fitness is defined by the density estimation method [39] that is adopted to truncate the individuals at the same front rather than the crowding distance in this study. The reason is that the crowding distance may mis-estimate the diversity of the population when the number of objectives comes up to more than two [40]. This mechanism is adopted to select individuals throughout this study. The output archive $\mathbb{FP}$ which stores feasible non-dominated individuals and the temporary archive $\mathbb{TA}$ which stores the best non-dominated individuals based on ICV are updated accordingly (Lines 21-25). Following the final update of $\mathbb{FP}$ in each generation, the reward R_t is calculated according to Equation (13) and $\mathbb{TA}$ in Section III-B3 (Lines 27-28). Besides, $\boldsymbol{s}_{t+1}$ is obtained according to Equation (12) (Line 29). Subsequently, $\mathbb{E}$ is updated with the newly generated experience sample $\mathbf{e}_t$ to facilitate later training and updating of the networks (Lines 30-32).

The proposed algorithm repeats the loop when the stopping criterion is not met. Otherwise, the feasible solutions in the output archive $\mathbb{FP}$ are taken as the Pareto optimal solutions for the given CMOP.

B. Computational time complexity

The time complexity of the proposed algorithm primarily comes from two components. The first component is the evolutionary process that mainly includes parent selection, population updating, and neighbor assignment. The second component is related to the deep Q-network, including the evaluation of the state, evaluation of the reward, and the training of the deep Q-network itself.

Suppose that the dimension of the objective space for CMOPs is m and the population size is N, the time complexity of parent selection in a generation is $O(mN^2)$. The time complexity of updating the population is determined by density estimation and truncation procedure [39], with the

worst-case complexity being $O(mN^3)$. The time complexity of the neighbor assignment is $O(mN^2)$, primarily determined by computing the angle between any two solutions.

In the deep Q-network, the computationally intensive task is to calculate the non-dominated rank of the population, which has a time complexity of $O(mN \log N)$ per generation. The complexity of calculating the reward is derived from computing either the IGD or the standard deviation of the population in the objective space, each with a computational complexity of $O(mN^2)$. Let l represent the number of hidden layers in the network, h represent the average number of nodes per hidden layer, r represent the number of nodes in the input layer, and o represent the number of nodes in the output layer. For training the network, each training session selects b samples for w epochs. The time complexity of training the deep Q-network is $O(((l-1)h^2+(r+o)h)wb)$ [10].

From the above analysis, the time complexity of the proposed algorithm for a maximum number of generations $T_{\max}$ is $O(T_{\max}(mN^3+((l-1)h^2+(r+o)h)wb)))$. While the deep Q-network training contributes to the overall time complexity, the integration does not change the computational order of either the evolutionary algorithm or the deep Q-network components.

V. EXPERIMENTAL SETUP

This section presents a series of experiments to validate the effectiveness of discovering systematic patterns in the temporal sequence of constraint handling selection within the proposed algorithm, denoted as CMOEA-TS. The experiments are given as follows:

- The first experiment is designed to directly validate the positive impact of discovering systematic patterns within the sequence for CMOEA-TS on solving CMOPs. By conducting a comparative analysis of CMOEA-TS against nine CMOEAs, we aim to demonstrate the enhanced capabilities of CMOEA-TS in terms of objective optimization and constraint satisfaction, as presented in Section V-B.
- The second experiment is designed to investigate how discovering the systematic patterns in the sequence based on the deep reinforcement learning influences the performance of CMOEA-TS, as presented in Section V-C.
- The third experiment is designed to contrast CMOEA-TS with a recent CMOEA based on the deep Q-network, as presented in Section V-D.
- The fourth experiment is designed to evaluate the influence of the selected CHTs on CMOEA-TS, as presented in Section V-E.
- The fifth experiment is designed to examine the effects of varying genetic operators on the performance of CMOEA-TS, as presented in Section V-F. In conjunction with the fourth experiment, the two experiments aim to validate that the comprehensive selection of both CHTs and genetic operators improves the adaptability and overall effectiveness of CMOEA-TS, enabling CMOEA-TS to dynamically respond to the generation-varying landscapes of a CMOP.
- The sixth experiment is designed to investigate the effectiveness of the proposed credit assignment function in generating the temporal sequence of constrained handling selection, as presented in Section V-G.
- Due to space limitations, the final experiment, which includes sensitivity analyses of the parameters (i.e., θ, ub, lb, μ_{thr}, λ, $T_{\max}$, N, and network structures), an analysis of the weaknesses of CMOEA-TS, a quantitative comparison of algorithm execution times, and a real-world application of CMOEA-TS to an unmanned aerial vehicle path planning problem [41], is presented in the supplementary materials.

A. Experimental setting

Three commonly used benchmark suites are adopted for the experimental studies, i.e., MW [42], LIR-CMOP [43], and DAS-CMOP [32]. The performance of CMOEA-TS is compared with nine algorithms, including MTCMO [6], cDPEA [44], BiCo [14], ShiP [9], CMOEA-MS [8], CCMO [45], PPS [19], CTAEA [12], and CMODQLMT [27]. The parameters related to CMOEA-TS and the compared algorithms are set as follows:

- Population Size: 100 for two-objective CMOPs and 300 for three-objective CMOPs.
- Termination Condition: For MW, the maximum generation $T_{\max}$ is set to 1000. For LIR-CMOP and DAS-CMOP, $T_{\max}$ is set to 1500. All compared algorithms share the same maximum number of function evaluations for each benchmark problem.
- Number of Independent Runs: All algorithms are run independently on each benchmark problem 30 times.
- Genetic Operators: For SBX, the crossover probability and distribution index are set to 1.0 and 20, respectively. For DE/rand/1/bin, the scalar factor (F) and crossover rate (CR) are set to 0.5 and 1.0, respectively. For DE/rand/2/bin, F and CR are set to 0.1 and 1.0, respectively. The mutation probability and distribution index of the polynomial mutation in the above genetic operators are set to $1/c$ and 20, respectively, as in the source reference [19].
- Parameters of Algorithms: The compared algorithms are implemented on PlatEMO 4.2 [46], using the platform's default parameters. For CMOEA-TS, the parameters are set as follows. θ, lb, ub, and μ_{thr} are set to $\pi/20$, -20, 20, and 0.1, respectively. For the deep Q-network, the number of nodes in the input layer r is set to the dimension of $\boldsymbol{s}_t$ and the number of nodes in the output layer o is set to 9. The number of hidden layers l is set to 5 and their numbers of nodes are 8, 16, 32, 16, and 8, respectively. The nodes of each layer are fully connected to each other using Exponential Linear Units (ELU) as the activation function. The learning rate of network training lr is set to 0.001. The update frequency of the target network is set to 10. The cardinality of $\mathbb{E}$ is set to 50. The parameters λ, ξ, b, and w are set to 0.1, 0.1, 32, and 10, respectively.

Two commonly used metrics, i.e., IGD and HV, are used to evaluate the performance of the CMOEAs in finding a set of Pareto optimal solutions. IGD mainly measures convergence by calculating the minimum sum of distances from points on

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2025.3584207

the true Pareto front to the set of solutions obtained by an algorithm. A smaller IGD value indicates better overall performance in terms of convergence. HV measures the volume enclosed by the nondominated solutions and a reference point, where each objective value of the reference point is set to 1.1 times the corresponding extreme objective value of the Pareto front in this paper. A larger HV value corresponds to better overall performance in terms of convergence and diversity. In addition, the Wilcoxon rank-sum test with a significance level of 0.05 and Friedman test on IGD results are used to analyze the results.

B. Performance of the proposed algorithm

Table II presents the comparisons of CMOEA-TS with the competing algorithms based on the IGD and Wilcoxon rank-sum test results on the MW, LIR-CMOP, and DAS-CMOP test instances. Additionally, the HV results obtained by all considered algorithms are provided in Table S-IV of the supplementary materials.

CMOEA-TS outperforms nine competing algorithms in the three benchmarks, i.e., MW, LIR-CMOP, and DAS-CMOP test instances, affirming its superiority in both IGD and HV metrics. Specifically, CMOEA-TS achieves the best average IGD values in 10 out of 14 MW test instances and attains the best HV values in 12 out of 14 instances, as detailed in Table S-IV. In the LIR-CMOP test instances, CMOEA-TS is the top performer in 9 out of 14 test instances according to both IGD and HV metrics. Moreover, CMOEA-TS demonstrates superior performance compared with the other algorithms in the DAS-CMOP test instances, with IGD values significantly better than the nine comparison algorithms in 6, 7, 7, 9, 5, 6, 9, 7, and 5 test instances, respectively. Similarly, the HV values are significantly better in 4, 8, 9, 9, 5, 6, 9, 9, and 7 test instances.

The average ranking of the Friedman statistical test is used to evaluate the performance of the algorithms, as shown in Fig. 5, where the lower ranking indicates better performance. The proposed algorithm has the best overall performance with the Friedman ranking of 2.25 among the compared algorithms. The result indicates that CMOEA-TS can find a set of better Pareto optimal solutions compared with the other algorithms on the three benchmark suites.

The enhanced performance of CMOEA-TS can be attributed to its strategic integration of the deep Q-network, aimed at identifying the consistent trends within the temporal sequence of constrained handling selection (see the analysis in Section V-C). The reinforcement learning agent in CMOEA-TS methodically adjusts the CHTs and genetic operators, thereby enhancing its responsiveness to balance feasibility, convergence, and diversity in each generation.

C. Effectiveness of discovering the systematic patterns within the sequence

CMOEA-TS is compared with its variant, denoted as TS-g. TS-g randomly selects action in each generation, while

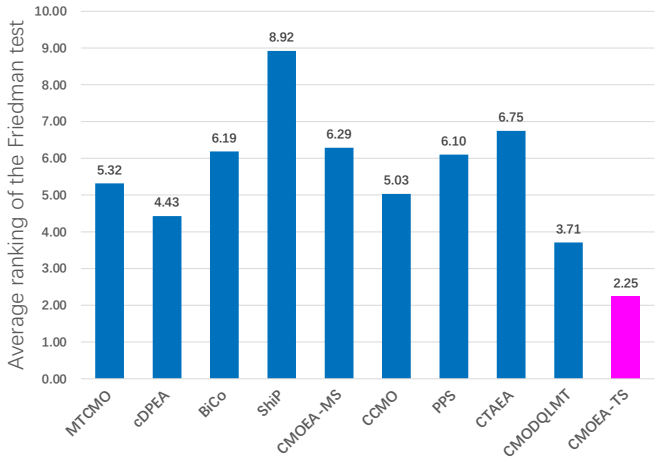


Fig. 5. Average ranking of the Friedman test for the ten compared CMOEAs on MW, LIR-CMOP, and DAS-CMOP test instances.

CMOEA-TS selects actions based on the deep Q-network. The rest of the parameter settings remains the same as in Section V-A. The Wilcoxon rank-sum test results are presented in Table III. The IGD and HV results are provided in Table S-V and Table S-VI in the supplementary materials, respectively. Additionally, Fig. 7(c) illustrates the comparative analysis based on the average ranking of the Friedman test.

The experimental results demonstrate that CMOEA-TS significantly outperforms TS-g on 26 of the 28 test problems in terms of HV, underscoring the crucial role of the deep Q-network in its operational strategy. CMOEA-TS methodically evaluates potential actions based on their historical success, thereby guiding CMOEA-TS to prioritize strategies that ensure an optimal balance of feasibility, convergence, and diversity. Consequently, the ability of CMOEA-TS to adapt its strategies based on both real-time feedback and accumulated insights directly enhances its performance and significantly bolsters its robustness in solving CMOPs.

An illustration is provided to offer deeper insight into the results, demonstrating how discovering the systematic patterns within the sequence correlates with the performance of CMOEA-TS on LIR-CMOP8, as shown in Figs. 6(a) through (f). Initially, the selections of constrained handling strategies in CMOEA-TS and TS-g exhibit varied responses, as shown in Fig. 6(a) and Fig. 6(b), indicative of an exploratory phase where various strategies are tested. Over time, these selections become more uniform, suggesting that CMOEA-TS has refined its selection process to consistently apply the most effective actions based on accumulated insights. As shown in Fig. 6(f) and Fig. 6(e), CMOEA-TS applies the effective actions that enable the population to traverse infeasible regions and reaches the Pareto front. In contrast, TS-g is impeded by the infeasible regions. This progression toward more consistent action choices enhances overall performance, as evidenced by the improved consistency of specific action indices throughout the evolutionary process. The impact of these refined selections across generations is further illustrated

TABLE II

IGD RESULTS OF CMOEA-TS AND THE OTHER NINE PEER ALGORITHMS ON THE THREE SETS OF BENCHMARK PROBLEMS. 'NAN' INDICATES THAT NO FEASIBLE SOLUTION IS FOUND IN AT LEAST ONE INSTANCE. '+', '-', AND '=' INDICATE RESULTS SIGNIFICANTLY BETTER, SIGNIFICANTLY WORSE, AND STATISTICALLY SIMILAR TO THAT OF CMOEA-TS, RESPECTIVELY. THE BEST AND SECOND-BEST AVERAGE VALUE ARE SHADED WITH DARK-GRAY AND A LIGHT-GRAY BACKGROUND, RESPECTIVELY.

Problem	MTCMO	cDPEA	BiCo	ShiP	CMOEA-MS	CCMO	PPS	CTAEA	CMODQLMT	CMOEA-TS
MW1	1.6141e-03 -	1.7386e-03 -	1.5508e-03 -	NAN -	NAN -	1.5167e-03 -	NAN -	2.0059e-03 -	1.7548e-03 -	1.4187e-03
MW2	2.0167e-02 -	1.7777e-02 -	1.2822e-02 -	2.7690e-02 -	1.9505e-02 -	2.1144e-02 -	NAN -	1.7420e-02 -	1.1253e-02 -	3.5662e-03
MW3	4.9790e-03 -	4.9408e-03 -	5.0196e-03 -	6.9993e-02 -	5.3174e-03 -	4.8433e-03 -	6.0906e-03 -	4.9782e-03 -	4.8177e-03 -	4.4840e-03
MW4	2.0992e-02 -	2.0436e-02 -	2.1131e-02 -	3.1903e-02 -	2.2800e-02 -	2.0999e-02 -	3.3616e-02 -	3.1317e-02 -	2.2674e-02 -	1.9735e-02
MW5	3.0878e-02 +	3.2740e-03 +	7.6446e-03 =	2.9643e-01 -	NAN -	1.4889e-03 +	5.0081e-01 -	1.2946e-02 -	1.4657e-03 =	1.9089e-03
MW6	2.2396e-02 -	1.1777e-02 -	8.9192e-03 -	4.3428e-02 -	2.4177e-02 -	5.4458e-02 -	NAN -	1.1188e-02 -	6.8106e-03 -	3.0439e-03
MW7	4.2417e-03 =	4.1677e-03 +	4.6564e-03 -	6.5628e-02 -	3.1032e-02 -	4.4736e-03 -	4.7087e-03 -	6.6625e-03 -	4.6033e-03 -	4.3028e-03
MW8	2.2008e-02 -	2.1227e-02 -	2.1790e-02 -	3.1922e-02 -	2.1854e-02 -	2.1506e-02 -	NAN -	2.8338e-02 -	2.1042e-02 -	2.0597e-02
MW9	5.8335e-03 -	2.9996e-02 =	8.5361e-03 -	6.0640e-02 -	1.1855e-01 -	5.2389e-03 -	1.0134e-01 -	8.7984e-03 -	8.3719e-03 -	5.6427e-03
MW10	4.2368e-02 -	2.4486e-02 -	3.1562e-02 -	NAN -	4.5238e-02 -	3.8625e-02 -	NAN -	1.6080e-02 -	1.0135e-02 -	3.5593e-03
MW11	1.5624e-02 =	1.5762e-02 =	1.5617e-02 =	3.3727e-01 -	3.3533e-02 =	1.5569e-02 =	1.6765e-02 -	2.3046e-02 -	1.5536e-02 =	1.5585e-02
MW12	1.3736e-02 =	2.7218e-02 -	4.7441e-03 -	3.4194e-02 -	1.8768e-02 -	4.9737e-03 -	1.5799e-01 -	7.9152e-03 -	4.8900e-03 -	4.6805e-03
MW13	6.9376e-02 -	2.9605e-02 -	2.7500e-02 -	2.6295e-01 -	9.5804e-02 -	5.8974e-02 -	5.0053e-01 -	4.4946e-02 -	1.9664e-02 -	1.0787e-02
MW14	5.3709e-02 -	5.3093e-02 -	5.5182e-02 -	7.4753e-02 -	6.7257e-02 -	5.3469e-02 -	8.1877e-02 -	5.4076e-02 -	5.4820e-02 -	5.0816e-02
LIR-CMOP1	1.3673e-01 -	1.5374e-01 -	2.0357e-01 -	NAN -	3.2864e-01 -	2.5401e-01 -	1.9875e-02 -	NAN -	1.4539e-01 -	1.6384e-02
LIR-CMOP2	1.1957e-01 -	1.5130e-01 -	1.7745e-01 -	NAN -	2.7794e-01 -	2.1375e-01 -	1.1329e-02 -	1.8830e-01 -	9.0760e-02 -	6.0173e-03
LIR-CMOP3	1.6111e-01 -	1.6851e-01 -	2.2735e-01 -	NAN -	3.2138e-01 -	2.7408e-01 -	2.1715e-02 +	NAN -	1.6421e-01 -	4.6944e-02
LIR-CMOP4	1.6057e-01 -	1.6994e-01 -	2.2653e-01 -	NAN -	3.2819e-01 -	2.9414e-01 -	8.5107e-03 +	NAN -	1.9777e-01 -	2.8008e-02
LIR-CMOP5	7.1157e-01 -	2.8865e-01 -	1.2201e+00 -	1.2183e+00 -	3.2029e-01 -	2.8229e-01 -	6.9498e-03 -	1.1289e+00 -	2.3324e-01 -	4.6912e-03
LIR-CMOP6	7.4372e-01 -	3.7924e-01 -	1.3463e+00 -	1.3467e+00 -	3.5149e-01 -	3.2036e-01 -	7.8179e-03 -	1.3149e+00 -	2.5338e-01 -	4.4317e-03
LIR-CMOP7	1.2900e-01 -	1.2648e-01 -	5.4894e-01 -	4.0054e-01 -	1.3143e-01 -	1.2001e-01 -	3.1361e-02 -	1.3696e-01 -	3.8183e-02 -	1.1802e-02
LIR-CMOP8	1.9243e-01 -	1.6992e-01 -	9.5949e-01 -	1.1099e+00 -	2.0579e-01 -	1.8321e-01 -	1.0908e-02 -	6.9988e-01 -	2.4374e-02 -	6.6122e-03
LIR-CMOP9	6.7425e-01 -	3.6354e-01 -	9.0014e-01 -	9.6509e-01 -	6.1746e-01 -	4.3234e-01 -	2.4566e-01 -	4.2761e-01 -	1.6932e-01 -	1.1749e-01
LIR-CMOP10	4.2444e-01 -	2.3711e-01 =	8.9992e-01 -	8.5083e-01 -	3.3323e-01 -	1.1715e-01 +	4.0801e-02 =	2.7816e-01 -	5.3252e-02 +	1.8957e-01
LIR-CMOP11	3.0857e-01 -	8.4994e-02 -	5.8145e-01 -	7.8407e-01 -	2.3167e-01 -	6.2393e-02 -	1.7335e-01 -	2.1036e-01 -	2.8844e-02 -	8.4163e-02
LIR-CMOP12	3.6769e-01 -	1.4996e-01 -	4.9759e-01 -	8.3405e-01 -	3.2646e-01 -	1.8613e-01 -	8.2681e-02 -	2.1431e-01 -	4.3202e-02 -	4.1441e-02
LIR-CMOP13	1.3036e+00 -	5.0749e-02 =	1.2628e+00 -	1.3086e+00 -	5.0123e-02 =	5.0198e-02 =	7.4207e-02 -	6.1719e-02 -	5.0562e-02 =	5.1095e-02
LIR-CMOP14	1.2598e+00 -	5.2679e-02 -	1.2605e+00 -	1.2648e+00 -	5.1808e-02 -	5.2637e-02 -	6.9681e-02 -	6.8121e-02 -	5.3007e-02 -	4.9824e-02
DAS-CMOP1	6.7678e-01 -	6.7199e-01 -	7.2262e-01 -	7.3497e-01 -	7.2503e-01 -	7.1577e-01 -	4.3293e-02 -	1.8143e-01 -	3.9358e-03 -	3.1340e-02
DAS-CMOP2	2.2439e-01 -	2.3009e-01 -	2.4778e-01 -	2.7325e-01 -	2.5727e-01 -	2.3026e-01 -	5.1108e-03 -	8.4193e-02 -	4.5949e-03 -	4.0280e-03
DAS-CMOP3	2.6231e-01 -	2.9566e-01 -	3.1033e-01 -	3.5220e-01 -	3.5157e-01 -	3.3981e-01 -	2.5048e-01 -	1.3241e-01 =	1.9858e-02 +	1.3395e-01
DAS-CMOP4	1.1598e-03 +	2.4504e-03 -	5.0753e-02 -	2.4131e-01 -	6.5908e-02 =	1.2633e-03 =	1.8297e-01 -	9.3364e-03 -	2.6901e-03 -	1.2782e-03
DAS-CMOP5	2.7042e-03 +	2.6552e-02 -	2.4655e-02 -	3.2127e-01 -	1.7754e-02 +	2.7445e-03 =	7.2893e-03 -	NAN -	2.9711e-03 -	2.7920e-03
DAS-CMOP6	4.0697e-02 =	2.1639e-02 =	8.0289e-02 -	4.7963e-01 -	1.5534e-01 -	3.3584e-02 -	NAN -	1.9486e-02 +	3.3460e-02 -	2.0759e-02
DAS-CMOP7	1.8209e-02 -	1.7948e-02 -	1.7861e-02 =	2.3190e-02 -	1.7516e-02 +	1.7928e-02 -	3.1470e-02 -	2.8820e-02 -	1.7828e-02 =	1.7762e-02
DAS-CMOP8	2.4689e-02 -	2.3249e-02 +	2.3469e-02 +	2.8422e-02 -	2.3731e-02 +	2.3993e-02 =	5.5317e-02 -	3.8468e-02 -	2.3116e-02 +	2.4142e-02
DAS-CMOP9	1.9211e-01 -	6.2221e-02 -	2.6720e-01 -	3.3909e-01 -	3.1294e-01 -	3.2284e-01 -	3.5977e-02 -	5.1977e-02 -	2.2874e-02 +	2.3411e-02
+/-/=	3/30/4	3/29/5	1/33/3	0/37/0	3/31/3	2/30/5	2/34/1	1/35/1	4/29/4	

in the convergence graph of the median HV value in Fig. 6(c) and Fig. 6(d), showing marked performance improvements compared with TS-g. It is noteworthy that TS-g achieves no improvement in HV, as it selects constrained handling strategies randomly, resulting in solutions that do not fall within the area enclosed by the reference point.

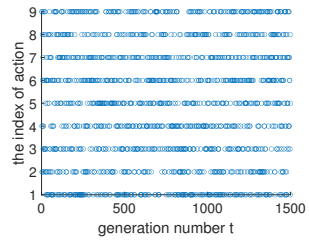
TABLE III

WILCOXON RANK-SUM TEST FOR IGD AND HV ON MW AND LIR-CMOP TEST INSTANCES.

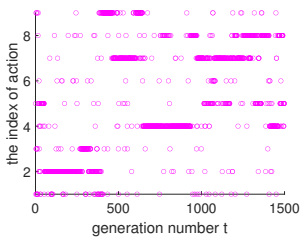
	IGD(+/-/=)	HV(+/-/=)
TS-g vs. CMOEA-TS	0/19/9	0/26/2

D. Comparison of the CMOEAs based on the deep Q-network

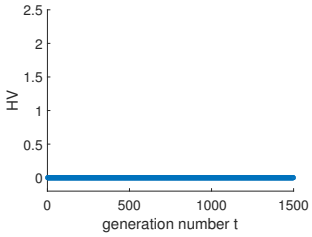
This subsection examines the performance of CMOEA-TS compared with another recent CMOEA that utilizes the deep Q-network, namely CMODQLMT. It is important to note that CMODQLMT primarily employs the deep Q-network to select auxiliary populations that assist in evolving solutions for CMOPs. By integrating the deep Q-network to systematically refine both the CHTs and genetic operators, CMOEA-TS more effectively adapts to the specific requirements of each generation. Consequently, CMOEA-TS achieves a better approximation of the Pareto front in the majority of the test instances, specifically in 33 out of 37 instances regarding the IGD metric, showing results no worse than those of CMODQLMT, as detailed in Table II.



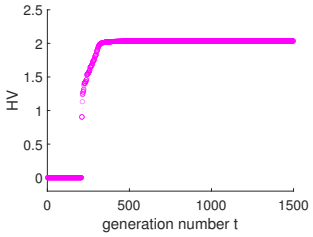
(a) Random constrained handling selections during the evolutionary process in TS-g.



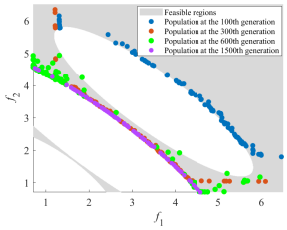
(b) Sequence of constrained handling selection during the evolutionary process in CMOEA-TS.



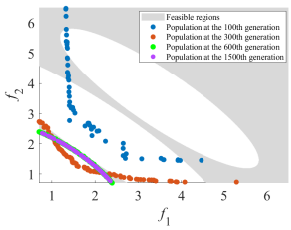
(c) Convergence graph obtained by TS-g based on the median HV value.



(d) Convergence graph obtained by CMOEA-TS based on the median HV value.



(e) Evolutionary process of TS-g on solving LIR-CMOP8.



(f) Evolutionary process of CMOEA-TS on solving LIR-CMOP8.

Fig. 6. Effectiveness of discovering the systematic patterns within the temporal sequence of constrained handling selection on LIR-CMOP8.

E. Influence of the selections of CHTs

CMOEA-TS is compared with its three variants: TS-a, TS-b, and TS-c. Each variant utilizes a distinct CHT: TS-a uses ICV, TS-b employs the ε-method, and TS-c utilizes CDP. In contrast, CMOEA-TS automatically selects one of these three CHTs based on the deep Q-network. The rest of the parameter settings remains the same as in Section V-A. CMOEA-TS, TS-a, TS-b, and TS-c are independently run 30 times on the MW and LIR-CMOP test instances. The Wilcoxon rank-sum test results are presented in Table IV. The IGD and HV results are provided in Table S-V and Table S-VI in the supplementary materials, respectively. Additionally, Fig. 7(a) illustrates the comparative analysis based on the average ranking of the Friedman test.

The results suggest that CMOEA-TS achieves the best performance, significantly outperforming TS-a and TS-c. The reason is that TS-a and TS-c prioritize constraint satisfaction and objective minimization, respectively, and such a bias is not beneficial for solving CMOPs. Although CMOEA-TS shows slightly inferior performance to TS-b in the LIR-CMOP1-4, it outperforms TS-b in the majority of the other test instances. TS-b adopts a dynamic decreasing constraint boundary approach, while CMOEA-TS can flexibly choose different CHTs according to the evolving requirements for balancing feasibility, convergence, and diversity during the evolutionary process. This flexibility effectively compensates for the deficiencies of a single CHT, thereby exhibiting enhanced robustness.

TABLE IV
WILCOXON RANK-SUM TEST FOR IGD AND HV ON MW AND LIR-CMOP TEST INSTANCES.

	IGD(+/-/=)	HV(+/-/=)
TS-a vs. CMOEA-TS	2/13/13	2/21/5
TS-b vs. CMOEA-TS	6/10/12	5/11/12
TS-c vs. CMOEA-TS	1/19/8	0/22/6

F. Influence of the selections of genetic operators

CMOEA-TS is compared with its three variants, denoted as TS-d, TS-e, and TS-f, each employing a distinct genetic operator: TS-d uses DE/rand/1/bin, TS-e deploys DE/rand/2/bin, and TS-f applies SBX. In contrast, CMOEA-TS automatically selects one of these three genetic operators based on the deep Q-network. The rest of the parameter settings remains the same as in Section V-A. The four algorithms are independently run 30 times on the MW and LIR-CMOP test instances. The Wilcoxon rank-sum test results are presented in Table V. The IGD and HV results are provided in Table S-V and Table S-VI in the supplementary materials, respectively. Additionally, Fig. 7(b) illustrates the comparative analysis based on the average ranking of the Friedman test.

The results show that CMOEA-TS achieves the best overall performance. The reason is that a single genetic operator cannot be applied to solve all problems, as shown in Table S-V. The algorithm with SBX significantly outperforms the algorithms with DE/rand/1/bin and DE/rand/2/bin on MW test instances, while the two DE genetic operators are more suitable than SBX for the CMOEA to solve LIR-CMOP test instances. The algorithms with DE/rand/1/bin and DE/rand/2/bin display variant performance on solving LIR-CMOP, hinting at their unique strengths. By integrating the three genetic operators, CMOEA-TS leverages their collective strengths and adaptively selects the most suitable genetic operator at different evolutionary stages when solving CMOPs.

G. Effectiveness of the credit assignment function in generating the sequence

This study designs an adaptive credit assignment function that evaluates each temporal sequence of constrained handling selection by assessing the executed actions at every generation. CMOEA-TS adjusts its strategy based on the rewards received from this function. To verify the effectiveness of the proposed

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2025.3584207

TABLE V
WILCOXON RANK-SUM TEST FOR IGD AND HV ON MW AND LIR-CMOP TEST INSTANCES.

	IGD(+/-/=)	HV(+/-/=)
TS-d vs. CMOEA-TS	5/18/5	3/20/5
TS-e vs. CMOEA-TS	1/21/6	0/23/5
TS-f vs. CMOEA-TS	0/19/9	4/21/3

credit assignment function, CMOEA-TS is compared with its variant, TS-h, where the rewards of TS-h are determined solely based on the improvement in IGD between two successive populations, as defined in Equation (13). The rest of the parameter settings remains the same as in Section V-A. Both algorithms are independently run 30 times on the MW and LIR-CMOP test instances. The Wilcoxon rank-sum test results are presented in Table VI. The IGD and HV results are provided in Table S-V and Table S-VI in the supplementary materials, respectively. Additionally, Fig. 7(d) illustrates the comparative analysis based on the average ranking of the Friedman test.

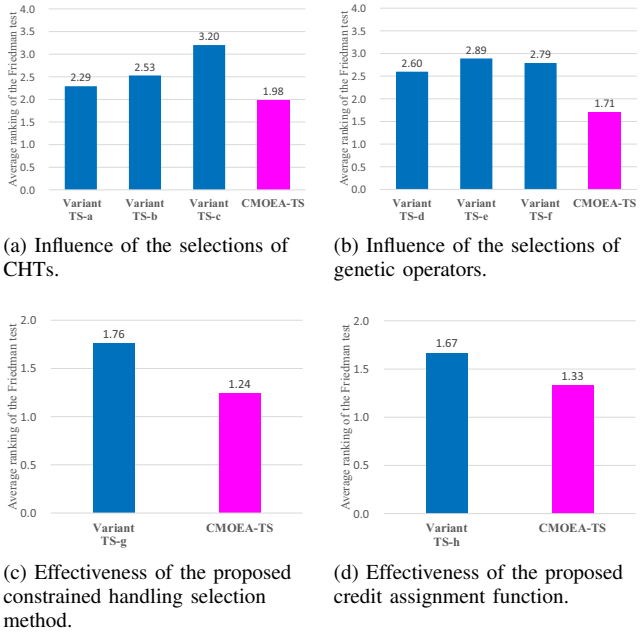
The results indicate that CMOEA-TS significantly outperforms TS-h. The underlying reason is that the reward scheme in TS-h is designed only to enhance objective optimization, without sufficiently considering population feasibility. This oversight leads the deep Q-network to learn suboptimal behaviors. In contrast, the reward scheme in CMOEA-TS is adaptively adjusted across different evolutionary stages, enabling a more comprehensive evaluation of strategy effectiveness. Consequently, the adaptive credit assignment function enables CMOEA-TS to achieve superior performance in obtaining Pareto optimal solutions compared with TS-h.

TABLE VI
WILCOXON RANK-SUM TEST FOR IGD AND HV ON MW AND LIR-CMOP TEST INSTANCES.

	IGD(+/-/=)	HV(+/-/=)
TS-h vs. CMOEA-TS	3/18/7	2/20/6

VI. CONCLUSION

This study focused on the temporal sequence of constrained handling selection for solving CMOPs. The selections of CHTs and genetic operators throughout the evolutionary process are modeled as the temporal sequence of constrained handling selection (see **Definition 1**). With this model, most CMOEAs focusing on the design of CHTs and genetic operators can be viewed as special cases. An essential aspect of effectively solving CMOPs with this sequence is utilizing historical data from past selections to predict upcoming ones, thereby enhancing decision-making capabilities. To achieve this, the deep Q-network was employed to identify systematic patterns within the sequence. It tracks and analyzes the effectiveness of various CHTs and genetic operators across successive generations. By pinpointing strategies that have yielded optimal outcomes in past generations, the proposed CMOEA dynamically adjusts its constrained handling strategies to effectively respond to the evolving dynamics of CMOPs.



(a) Influence of the selections of CHTs.

(b) Influence of the selections of genetic operators.

(c) Effectiveness of the proposed constrained handling selection method.

(d) Effectiveness of the proposed credit assignment function.

Fig. 7. Average ranking of the Friedman test for CMOEA-TS and its variants on MW and LIR-CMOP test instances.

The statistical experimental results from the Wilcoxon rank-sum test and Friedman test showed that CMOEA-TS based on the temporal sequence of constrained handling selection outperformed the nine competing CMOEAs. These results verify that discovering the systematic patterns within the sequence for CMOEA-TS has a positive impact on solving CMOPs in terms of objective optimization and constraint satisfaction, as presented in Section V-B. Additionally, the effectiveness of the deep Q-network in recognizing consistent trends in the temporal sequence of constrained handling selection was validated in Section V-C. This network empowers CMOEA-TS to dynamically adapt its strategies to meet the requirements of balancing feasibility, convergence, and diversity throughout the evolutionary process. Moreover, the two experiments validated that the comprehensive selection of both CHTs and genetic operators improves the adaptability and overall effectiveness of CMOEA-TS in solving CMOPs, as presented as in Section V-E and Section V-F.

In the future, we plan to extend the applicability of CMOEA-TS to large-scale constrained optimization problems, focusing on its adaptability and performance with an increased number of decision variables and constraints. Furthermore, we aim to adapt CMOEA-TS for dynamic constrained optimization problems, emphasizing scalability and efficiency in computationally intensive scenarios, particularly where constraints and objectives are subject to change over time.

REFERENCES

[1] G. Peng, H. Wu, H. Wu, and K. Wolter, "Constrained multiobjective optimization for IoT-enabled computation offloading in collaborative edge and cloud computing," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 723–13 736, 2021.

[2] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. P. Chen, "Multiobjective overtaking maneuver planning of autonomous ground vehicles," *IEEE Transactions on Cybernetics*, vol. 51, no. 8, pp. 4035–4049, 2021.

[3] S. Yin, R. Wang, J. Zhang, and Y. Wang, "Asynchronous parallel expected improvement matrix-based constrained multi-objective optimization for analog circuit sizing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 9, pp. 3869–3873, 2022.

[4] Y. Hou, Y. Zhang, N. Wu, and Q. Zhu, "Constrained multi-objective optimization of short-term crude oil scheduling with dual pipelines and charging tank maintenance requirement," *Information Sciences*, vol. 588, pp. 381–404, 2022.

[5] 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 multi-objective problems with multi-constraint," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 267–280, 2024.

[6] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, C. Yue, H. Lin, and K. C. Tan, "Dynamic auxiliary task-based evolutionary multitasking for constrained multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 642–656, 2023.

[7] 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, 2023.

[8] 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, 2022.

[9] Z. Ma and Y. Wang, "Shift-based penalty for evolutionary constrained multiobjective optimization and its application," *IEEE Transactions on Cybernetics*, vol. 53, no. 1, pp. 18–30, 2023.

[10] M. Zuo, D. Gong, Y. Wang, X. Ye, B. Zeng, and F. Meng, "Process knowledge-guided autonomous evolutionary optimization for constrained multiobjective problems," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 193–207, 2024.

[11] J. Wang, Y. Li, Q. Zhang, Z. Zhang, and S. Gao, "Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 6, pp. 3476–3491, 2022.

[12] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2019.

[13] F. Ming, W. Gong, L. Wang, and L. Gao, "Constrained multi-objective optimization via multitasking and knowledge transfer," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 77–89, 2024.

[14] Z.-Z. Liu, B.-C. Wang, and K. Tang, "Handling constrained multiobjective optimization problems via bidirectional coevolution," *IEEE Transactions on Cybernetics*, vol. 52, no. 10, pp. 10 163–10 176, 2022.

[15] F. Ming, W. Gong, D. Li, L. Wang, and L. Gao, "A competitive and cooperative swarm optimizer for constrained multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1313–1326, 2023.

[16] M. Ming, A. Trivedi, R. Wang, D. Srinivasan, and T. Zhang, "A dual-population-based evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 739–753, 2021.

[17] Z.-Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 870–884, 2019.

[18] Z. Zhang, Z. Hao, and H. Huang, "Hybrid swarm-based optimization algorithm of GA & VNS for nurse scheduling problem," in *Information Computing and Applications: Second International Conference, ICICA 2011, Qinhuangdao, China, October 28-31, 2011. Proceedings 2*. Springer, 2011, pp. 375–382.

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

[20] R. Sun, J. Zou, Y. Liu, S. Yang, and J. Zheng, "A multi-stage algorithm for solving multi-objective optimization problems with multi-constraints," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1207–1219, 2023.

[21] K. Zhang, Z. Xu, G. G. Yen, and L. Zhang, "Two-stage multi-objective evolution strategy for constrained multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 28, no. 1, pp. 17–31, 2024.

[22] Y. Xiang, X. Yang, H. Huang, and J. Wang, "Balancing constraints and objectives by considering problem types in constrained multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 51, no. 1, pp. 88–101, 2023.

[23] C. Peng, H.-L. Liu, and E. D. Goodman, "A cooperative evolutionary framework based on an improved version of directed weight vectors for constrained multiobjective optimization with deceptive constraints," *IEEE Transactions on Cybernetics*, vol. 51, no. 11, pp. 5546–5558, 2021.

[24] Z. Ma, Y. Wang, and W. Song, "A new fitness function with two rankings for evolutionary constrained multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 5005–5016, 2021.

[25] K. Qiao, K. Yu, B. Qu, J. Liang, C. Yue, and X. Ban, "Feature extraction for recommendation of constrained multi-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 4, pp. 949–963, 2023.

[26] C. Peng, H.-L. Liu, E. D. Goodman, and K. C. Tan, "A two-phase framework of locating the reference point for decomposition-based constrained multi-objective evolutionary algorithms," *Knowledge-Based Systems*, vol. 239, p. 107933, 2022.

[27] F. Ming, W. Gong, and L. Gao, "Adaptive auxiliary task selection for multitasking-assisted constrained multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 18, no. 2, pp. 18–30, 2023.

[28] H. M. Maldonado and S. Zapotecas-Martínez, "A dynamic penalty function within MOEA/D for constrained multi-objective optimization problems," in *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2021, pp. 1470–1477.

[29] A. dos Santos Mignon and R. L. d. A. da Rocha, "An adaptive implementation of ε-greedy in reinforcement learning," *Procedia Computer Science*, vol. 109, pp. 1146–1151, 2017.

[30] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[31] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, "A survey on evolutionary constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 2, pp. 201–221, 2023.

[32] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evolutionary Computation*, vol. 28, no. 3, pp. 339–378, 2020.

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

[34] K. Deb, R. B. Agrawal *et al.*, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 2, pp. 115–148, 1995.

[35] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[36] M. Haldar, M. Abdool, P. Ramanathan, T. Xu, S. Yang, H. Duan, Q. Zhang, N. Barrow-Williams, B. C. Turnbull, B. M. Collins *et al.*, "Applying deep learning to airbnb search," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1927–1935.

[37] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[38] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[39] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[40] S. Kukkonen and K. Deb, "A fast and effective method for pruning of non-dominated solutions in many-objective problems," in *International Conference on Parallel Problem Solving from Nature*. Springer, 2006, pp. 553–562.

[41] C. Peng and S. Qiu, "A decomposition-based constrained multi-objective evolutionary algorithm with a local infeasibility utilization mechanism for UAV path planning," *Applied Soft Computing*, vol. 118, p. 108495, 2022.

[42] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.

[43] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in MOEA/D for cmops with large infeasible regions," *Soft Computing*, vol. 23, pp. 12 491–12 510, 2019.

[44] C. Wang, Z. Wang, Y. Tian, X. Zhang, and J. Xiao, "A dual-population based evolutionary algorithm for multi-objective location problem under uncertainty of facilities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7692–7707, 2022.

[45] 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, 2021.

[46] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.