ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

Constrained Multi-objective Differential Evolutionary Algorithm with Adaptive Constraint Handling Technique

Caixia Cui[1, a, *], Qinqin Fan[1]

[1]School of Logistics Science and Engineering Research Institute, Shanghai Maritime University, Shanghai 201306, China

[a]Corresponding author e-mail: 1071509289@qq.com

Abstract

Finding feasible solutions and a good approximate Pareto front (PF) are two important tasks in the constrained multi-objective optimization (CMO). Various constraint handling techniques (CHTs) have a significant impact on these two tasks. To realize the adaptive adjustment of CHTs, a constrained multi-objective differential evolution algorithm with adaptive constraint handling technique (ACHT-CMODE) is proposed in the current study. In the ACHT-CMODE, three state-of-the-art constraint handling methods are integrated and an improved reverse generation distance is used to evaluate their performances. Also, the Q-learning method is utilized to guide the evolution of CHTs. The performance of the ACHT-CMODE is compared with that of the other five constraint multi-objective evolutionary algorithms on 18 test functions. Experimental results show that the overall performance of the ACHT-CMODE is the best among all compared algorithms, and the proposed algorithm is capable of selecting a suitable constraint handling method to solve a particular type of constrained multi-objective optimization problems (CMOPs).

Keywords

Constrained multi-objective optimization; Constraint handling technique; Adaptive; Q-learning.

1. INTRODUCTION

Constrained multi-objective optimization problems (CMOPs) widely exist in scientific research and engineering practice, such as multi-objective constrained trajectory optimization problems [1], constrained multi-objective robot design [2], energy and reserve procurement design [3], etc. Without loss of generality, CMOPs can be formally expressed as [4]:

$$\begin{aligned} \min\ & f(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x})\right)^T \\ \text{s.t.}\ & g_j(\boldsymbol{x}) \le 0, j = 1, 2, \cdots, p \\ & h_j(\boldsymbol{x}) = 0, j = p+1, \cdots, q \\ & \boldsymbol{x} = \left(x_1, x_2, \cdots, x_D\right)^T \in O \end{aligned} \tag{1}$$

Where f(x) is the objective function, which includes m objectives; x is a D-dimensional decision vector, O is the decision space; $g_j(\boldsymbol{x})$ is the j-th inequality constraint; $h_j(\boldsymbol{x})$ is the (j-p)-th equality constraint; p is the number of inequality constraints, thus the number of equality constraints is q-p.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

To solve CMOPs, various constrained multi-objective evolutionary algorithms (CMOEAs) have been proposed. For example, Jan and Zhang [5] introduced a novel method to adjust the evolution of infeasible solutions, which penalizes infeasible solutions via an adaptive threshold value. Fan et al. [6] proposed an angle-based constrained-domination principle (ACDP) to solve CMOPs. Datta et al. [7] put forward a HyCon algorithm, which combines multi-objective optimization method and penalty function method. In [8], a new push and pull search framework is proposed, which divides the search process into two stages. In the push stage, non-dominated solutions are preferred with ignoring the constraints; in the pull stage, an improved epsilon constraint-handling method is used.

Based on the above introductions, improving the performance of a single CHT or mixing multiple CHTs to solve CMOPs commonly seen in the CMO. Inspired by studies [9], [10], a constrained multi-objective differential evolution algorithm with adaptive constraint handling technique (ACHT-CMODE) is introduced in the current study. ACHT-CMODE uses the Q-learning method [11] to self-adaptively select a suitable CHT in the next iteration.

The main contributions of this work are highlighted as follows.

(1) An adaptive constraint handling technique is proposed. Three different types of CHTs are integrated into the proposed method, and an appropriate CHT can be automatically selected during the entire evolutionary process. Therefore, the proposed method can integrate advantages of different CHTs.

(2) The proposed method is incorporated into a multi-objective differential evolution algorithm to solve CMOPs. Systematic experiments demonstrate that the ACHT-CMODE provides high-quality solutions on 18 widely used benchmark test functions.

The rest of this paper is organized as follows. Section 2 and 3 briefly introduce some basic concepts and related studies on constrained multi-objective optimization, respectively. The details of the proposed algorithm are given in Section 4. Subsequently, the experiments results and discussions are presented in Section 5. Finally, Section 6 concludes this work.

2. BASIC CONCEPTS

2.1. Basic Definitions in CMO

In this subsection, some terminologies and concepts of the CMO are descripted as follows [12], [13], [14]:

1) *Pareto Dominance*: The condition that vector $\boldsymbol{u}$ dominates another vector $\boldsymbol{v}$ (denoted as $\boldsymbol{u} \succ \boldsymbol{v}$) is: iff $\forall n \in \{1, 2, \ldots m\}$, $u_n \le v_n$ and $\boldsymbol{u} \ne \boldsymbol{v}$.

2) *Pareto Optimal Set*: A vector $\boldsymbol{x}^* \in O$, if there is no other vector $\boldsymbol{x} \in O$ such that $f(\boldsymbol{x}) \succ f(\boldsymbol{x}^*)$, $\boldsymbol{x}^*$ is called Pareto solution. The set of all the Pareto optimal solutions (denoted as X^*) is called Pareto optimal set.

3) *Pareto Front*: For a given CMOP, the Pareto Front is defined as $PF = \left\{ f\left(\boldsymbol{x}^*\right) \mid \boldsymbol{x}^* \in \boldsymbol{X}^* \right\}$.

4) *Feasible Solution*

A solution $\boldsymbol{x}$ satisfying $C(\boldsymbol{x}) = 0$ is called a feasible solution, where $C(\boldsymbol{x}) = \sum_{j=1}^{q} C_j(\boldsymbol{x})$ is the degree of constraint violation of $\boldsymbol{x}$, and $C_j(x) = \begin{cases} \max\left(0, g_j(\boldsymbol{x})\right), & 1 \le j \le p \\ \max\left(0, \left|h_j(\boldsymbol{x}) - \delta\right|\right), & p+1 \le j \le q \end{cases}$, δ is the tolerance value of equality constraints and generally set as a small positive number.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

2.2. Constrained Handling Technique

Three constrained handling techniques are introduced as following.

1) Self-adaptive penalty function

Self-adaptive penalty function (SP) [15] has been widely used because of its simple implementation, which uses a new fitness value to select individuals and is defined as follows:

$$M_n(\boldsymbol{x})=l_n(\boldsymbol{x})+b_n(\boldsymbol{x}) \tag{2}$$

$$l_n(\boldsymbol{x})=\begin{cases} C(\boldsymbol{x}), & if\ r_f=0 \\ \sqrt{\tilde{f}_n(\boldsymbol{x})^2+\bar{C}(\boldsymbol{x})^2}, & \text{otherwise} \end{cases} \tag{3}$$

$$b_n(\boldsymbol{x})=\left(1-r_f\right)c_n(\boldsymbol{x})+r_f e_n(\boldsymbol{x}) \tag{4}$$

Where $l_n(\boldsymbol{x})$ is n-th distance value; $b_n(\boldsymbol{x})$ is n-th penalty value; $\tilde{f}_n(\boldsymbol{x})$ is the normalized objective function value; $\bar{C}(\boldsymbol{x})=\frac{1}{q}\sum_{j=1}^{q}\frac{C_j(\boldsymbol{x})}{C_j^{\max}}$; r_f is the feasible rate of current population; and $c_n(\boldsymbol{x})=\begin{cases} 0, & \text{if } r_f=0 \\ \bar{C}(\boldsymbol{x}), & \text{otherwise} \end{cases}$, $e_n(\boldsymbol{x})=\begin{cases} 0, & \text{if } x \text{ is a feasible individual} \\ \tilde{f}_n(\boldsymbol{x}), & \text{otherwise} \end{cases}$.

2) Constrained domination principle

Constrained domination principle (CDP) proposed by Deb [16] is one of the most famous methods, which is commonly used to select individuals:

For two feasible individuals, the Pareto non-dominant individual is preferred;

For one feasible individual and one infeasible individual, the former is preferred;

For two infeasible individuals, the one with less constraint violation degree is preferred.

3) Adaptive tradeoff model

Adaptive tradeoff model (ATM) [17] divides the whole constraint optimization process into three stages according to the feasible rate of current population P:

Infeasible stage. The degree of constraint violation is regarded as an additional objective. Then, individuals are layered by non-dominated sorting [16] and then half of the individuals with small constraint violation degrees in the first layer are stored in the offspring population, then delete them from population. The same operation is performed on remaining individuals until the number of offspring population reaches the population size

Semi-feasible stage. Similar to SP, the ATM uses a new function via the normalized objective function value and the constraint violation degree, which is calculated as follows:

$$M_n(\boldsymbol{x})=\tilde{f}_n(\boldsymbol{x})+\tilde{C}(\boldsymbol{x}) \tag{5}$$

$$\tilde{f}_n(\boldsymbol{x})=\frac{f_n'(\boldsymbol{x})-\min\limits_{\boldsymbol{x}\in P} f_n'(\boldsymbol{x})}{\max\limits_{\boldsymbol{x}\in P} f_n'(\boldsymbol{x})-\min\limits_{\boldsymbol{x}\in P} f_n'(\boldsymbol{x})} \tag{6}$$

$$\tilde{C}(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in P_Y \\ \dfrac{\bar{C}(\boldsymbol{x}) - \min\limits_{\boldsymbol{x} \in P_N} \bar{C}(\boldsymbol{x})}{\max\limits_{\boldsymbol{x} \in P_N} \bar{C}(\boldsymbol{x}) - \min\limits_{\boldsymbol{x} \in P_N} \bar{C}(\boldsymbol{x})}, & \boldsymbol{x} \in P_N \end{cases} \tag{7}$$

Where $f_n^{'}(\boldsymbol{x}) = \begin{cases} f_n(\boldsymbol{x}), & \boldsymbol{x} \in P_Y \\ \max\{\varphi * f_{\min} + (1-\varphi) * f_{\min}\}, & \boldsymbol{x} \in P_N \end{cases}$ and φ represents the feasible rate of the last iteration population; P_Y and P_N represents the set of feasible and infeasible solutions in P, respectively.

Feasible stage. Non-dominated sorting method is used to select individuals.

2.3. Performance Metrics

In our experiments, three performance metrics are used to evaluate the performance of all compared algorithms.

1) Inverted Generational Distance (IGD)

IGD [18] is defined as follows:

$$IGD(H, PF^*) = \frac{\sqrt{\sum_{z^* \in PF^*} d(z^*, H)^2}}{|PF^*|} \tag{8}$$

Where H is the approximation of PF; PF* represents a set of solutions uniformly distributed along the real PF; d(z*,H) represents Euclidean distance between z^* and H nearest point; |PF*| denotes the number of solutions in PF*. Small IGD value means an algorithm can find a good approximate PF.

2) Hypervolume (HV)

HV [19] is calculated as:

$$HV(H) = L\left(\bigcup_{z \in H} [z_1, z_1^r] \times \cdots \times [z_m, z_m^r]\right) \tag{9}$$

Where L is the Lebesgue measure; $\boldsymbol{z}=(z_1, \ldots, z_m)$ denotes a solution in H; and $\boldsymbol{z}^r = (z_1^r, \ldots, z_m^r)$ represents a worst point dominated by all the Pareto optimal solutions.

3) Feasible Rate (FR)

FR [4] is formulated as follows:

$$FR = \frac{FSR}{TTR} \tag{10}$$

Where FSR denotes the number of runs in which the algorithm can find at least one feasible solution in the final population; TTR denotes total run times. A large FR value means that an algorithm can find feasible solutions with a high probability.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

3. RELATED WORK

Meta-heuristic algorithms have many advantages when compared with traditional optimization methods, thus they are widely used to solve different types of optimization problems. Specifically, CMOPs, which are both constrained and multi-objective, belong to a particular type of multi-objective optimization problems. i.e., the CMO aims to find a good approximate PF and a set of feasible solutions. Therefore, CMOEAs combine CHTs and MOEAs are effective method to solve CMOPs.

Jadaan et al. [20] combined multi-objective genetic algorithm with a parameterless and adaptive penalty function to solve CMOPs, in which two ranks obtained by the roulette wheel selection are considered as penalty parameters. Qu et al. [21] proposed a diversity enhanced constrained multi-objective differential evolution algorithm (DE-CMODE). DE-CMODE mainly introduced an external memory to enhance the population diversity for alleviating the premature convergence problem. Based on NSGA-II [16] and modified differential evolution operators, Hsieh et al. [22] proposed a hybrid constraint handling mechanism, which integrates the penalty function method and the ε constraint method. In their proposed algorithm, if the constraint violation degree of individuals exceeds ε value, then the penalty function method is used. In [23], a novel multi-objective hybrid particle swarm algorithm was proposed, in which a simulated annealing with variable neighborhood structure and an external archive are used. Experimental results show that the proposed algorithm has good performance. Based on MOEA/D [13], Yang et al. [24] introduced ε constraint method and adaptive operator selection to tackle CMOPs. The results on 10 benchmark test functions indicate that the proposed method is effective. Yu et al. [25] designed a novel mutation mechanism based on different evolution (DE) to solve CMOPs, the DE mutation operator and the Gauss mutation operator are applied to deal with infeasible solutions and feasible solutions, respectively. Ming et al. [26] proposed a dual-grid dual-phase strategy (DPPS) to solve CMOPs on the basis of the MOEA/D. The DPPS used two populations to explore the feasible and infeasible regions separately in the push stage, and then exchanged useful information between two populations in the pull stage. Experimental results show that the proposed algorithm is competitive. In [27], the proposed algorithm divides the whole optimization process into two phases: the employed bee and the onlooker bee. The fast nondominated sorting and the crowded distance are employed in the former phase, and the Tchebycheff approach is used to improve diversity in the latter stage.

4. PROPOSED ALGORITHM

No CHT can perform best on all types of CMOPs due to the No Free Lunch [28]. To integrate the advantages of different CHTs, an adaptive constraint handling method is proposed in this work. In the proposed method, different common CHTs will be self-adaptively selected via the Q-learning. Moreover, the proposed method is incorporated into a multi-objective differential evolution algorithm, named as ACHT-CMODE. Its main operators are described in the following subsections.

4.1. Adaptive Constraint Handling Technique

To adapt different types of CMOPs, an adaptive constraint handling technique is proposed in this work. Because the Q-learning method is used to select a suitable CHT and SP, CDP, ATM are selected in the proposed algorithm, the action space A can be expressed as A = [SP, CDP, ATM] and the three states can be defined as S = [excellent, medium, poor]. Moreover, the value of reward R is [1, 0, -1] [29]. The form of Q-table is shown in Table 1.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

Table 1. The form of Q-table

Q-table	SP	CDP	ATM
excellent	Q (1,1)	Q (1,2)	Q (1,3)
medium	Q (2,1)	Q (2,2)	Q (2,3)
poor	Q (3,1)	Q (3,2)	Q (3,3)

The steps of ACHT are given in Algorithm 1, which mainly consists of three parts: 1) assess the actions; 2) update the Q-table; 3) use ε-greedy method to select the action.

1) Performance assessment

To assess the performance of three CHTs, an improved reverse generation distance is proposed as follows:

$$mIGD(H,\overline{PF}) = \frac{\sqrt{\sum_{\overline{z}\in\overline{PF}} d(\overline{z},H)^2}}{\left|\overline{PF}\right|} \tag{11}$$

Where $\overline{PF}$, is the approximation of PF, which is selected from all actions.

Each individual has its own CHT, thus the populations of three CHTs can be denoted as PSP, PCDP, PATM. Moreover, their approximate PF can be denoted as HSP, HCDP, HATM. According to the Eq. (11), mIGD values of three CHT can be achieved, i.e., mIGDSP, mIGDCDP, mIGDATM.

A small mIGD value means the action is better. The minimum mIGD value represents the individual chooses this CHT is in "excellent" state and its reward will be 1; the middle mIGD value represents the state of corresponding CHT is "medium" and the reward is 0; while the reward of "poor" CHT is -1.

2) Update Q-table.

Using a Q-table to store the Q(s,a) value is one of the main steps in the Q-learning method. Q-table can be updated as follows:

$$Q(s,a) = Q(s,a) + \alpha\left(r+\gamma Q(s',a')\text{-}Q(s,a)\right) \tag{12}$$

Where a (a∈A) represents the current action; s (s∈S) represents the current state; r (r∈R) represents the reward; α is the update step size; γ is the reward discount; s' (s'∈S) represents the state at the next moment; a' (a'∈A)denotes the best action in s'. The values of α and γ are set to be 0.5 and 0.8 [29] in the current study, respectively.

3) Action selection using the ε-greedy

Every action can be selected via the ε-greedy method, but the best action is more likely to be selected, and its selection probability is $1-\varepsilon+\frac{\varepsilon}{|\boldsymbol{A}|}$ [11] ($|\boldsymbol{A}|$ denotes the number of actions in $\boldsymbol{A}$). Moreover, remaining actions are selected with a uniform probability $\frac{\varepsilon}{|\boldsymbol{A}|}$. The value of ε is set to be 0.2.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

Algorithm 1: ACHT

Input: SV: the state vector

RC: the reward chain

NP: the population size

P_{SP}^{G}, P_{CDP}^{G}, P_{ATM}^{G} : the populations corresponding to the three CHTs under the G-th iteration

Q: state-action table

1 Calculate $\mathrm{mIGD}_{\mathrm{SP}}$, $\mathrm{mIGD}_{\mathrm{CDP}}$, and $\mathrm{mIGD}_{\mathrm{ATM}}$ according to formula (11);

2 **for** i =1: NP **do**

3 Obtain the s’ and r of i-th individual using mIGD value, then **SV** and **RC** are updated;

4 Update Q-table by formula (12);

5 **end for**

6 Generate action a by executing ε-greedy method and get the action chain **AC**;

7 **Output: AC**.

4.2. Overall Implementation of the Proposed Algorithm

There are three main operators in the proposed ACHT-CMODE, i.e., initialization, population evolution, and ACHT. The pseudocode of ACHT-CMODE is given in Algorithm 2. It can be observed from Algorithm 2 that, for the initialization operator (lines 1-6 in Algorithm 2), the initial population is divided into three subpopulations using the k-means clustering [11]. Their decision vectors are denoted as P_1, P_2, P_3 and objective vectors are named as H_1, H_2, H_3. Therefore, mIGD values of three subpopulations can be calculated by Eq. (11), i.e., mIGD_1, mIGD_2, mIGD_3.

For the population evolution (line 8-18), similar to the Ref. [30], the mutation strategy DE/current-to-rand/1 and the crossover strategy SBX [31] are used with a high probability in the early evolution stages, and the DE/rand-to-best/1/bin is more likely to be selected in the later phases of the population evolution.

For the ACHT, the Q-learning algorithm is mainly used to realize this process, and the specific steps are detailed in algorithm 1.

Finally, feasible solutions are choosed to enter in the external archive **B**.

Algorithm 2: Procedure of ACHT-CMODE

Input: NP: the population size

G: current iteration number

Gmax: the maximum number of iterations

F_{pool}: the pool of the scaling factor F, i.e. {0.6, 0.8, 1.0}

CR_{pool}: the pool of the crossover control parameter CR, i.e. {0.1, 0.2, 1.0}

$\boldsymbol{x}_i^G$, $\boldsymbol{v}_i^G$, $\boldsymbol{u}_i^G$: i-th target vector, mutant Vector, trial vector under the G-th iteration

$\boldsymbol{x}_{r_1}^G$, $\boldsymbol{x}_{r_2}^G$, $\boldsymbol{x}_{r_3}^G$: three different target vectors randomly selected in P^G

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

x_{best}^{G} : the individual with the best performance under the G-th iteration

$rand_j$: uniform random number between [0,1]

j_{rand}: random integer within the range of [1,D]

Initialize population $P^{G}=\left\{x_{1}^{G},\ldots,x_{i}^{G},\ldots x_{NP}^{G}\right\}$ and G=1;
Initialize the external archive **B,** Q-table, the state vector **SV,** the action chain **AC** and the reward chain **RC**;
Divide P^G into three parts by k-means clustering;
Calculate mIGD values of three parts by formular (11);
Judge individuals' state by mIGD values and **SV** is obtained;
The action chain **AC** is obtained by ε-greedy method;
for G =1:Gmax **do**
    for i =1: NP **do**
        Randomly select a F value from the F_{pool};
        Randomly select a CR value from the CR_{pool};
        if rand< G/Gmax **then**
            $v_{i}^{G}=x_{i}^{G}+rand*\left(x_{r_1}^{G}-x_{i}^{G}\right)+F*\left(x_{r_2}^{G}-x_{r_3}^{G}\right)$
            Generate u_{i}^{G} by applying the SBX;
        else
            $v_{i}^{G}=x_{r_1}^{G}+F*\left(x_{\text{best}}^{G}-x_{r_1}^{G}\right)+F*\left(x_{r_2}^{G}-x_{r_3}^{G}\right)$
            Generate u_{i}^{G} by applying the binomial crossover;
        end if
    end for
    Select individuals from [P^G; $\mathbf{u}^G$] to enter P_{SP}^{G}, P_{CDP}^{G} and P_{ATM}^{G} according to **AC;**
    Update **AC** according to Algorithm 1;
    P^{G+1}=[P_{SP}^{G}; P_{CDP}^{G}; P_{ATM}^{G}];
    Save the feasible solutions at the first level of non-dominated sorting in the population P^{G+1} to the external archive **B**;
end for
if |**B**|> NP, use the non-dominated sorting method to select NP individuals from **B** as final solution set P; otherwise, all feasible solutions in **B** enter P; and get corresponding objective vectors H.
Output: P, H.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

5. EXPERIMENTAL RESULTS AND DISCUSSIONS

To demonstrate the performance of the ACHT-CMODE, five CMOEAs are selected in the current study. Namely, SP-CMODE, CDP-CMODE, ATM-CMODE, TOP-NSGAII-CDP [4] and HypE-FR [32], in which the structures of SP-CMODE, CDP-CMODE, ATM-CMODE are consistent with ACHT-CMODE except the slight difference in CHT, using SP, CDP and ATM to select individuals respectively while ACHT-CMODE uses ACHT. Additionally, two non-parametric statistical test methods are utilized to analyze the search performances of all the compared algorithms, which are Wilcoxon's rank sum test [33] and Friedman's test [34]. "+", "-" and "≈" denote that the performance of ACHT-CMODE is better than, worse than, and similar to that of compared algorithms, respectively.

5.1. Test Instances

All experiments are conducted on NCTP [14], which can be divided into three types:1) Type I: NCTP-1~NCTP-6. The constrained PF is composed of a part of the boundary of the feasible region, which is discontinuous; 2) Type II: NCTP-7~NCTP-12. The constrained PF consists of a part of the unconstrained PF and a part of the boundary of the feasible region, containing continuous and discontinuous parts; 3) type III: NCTP-13~NCTP-18. The constrained PF is the same with the unconstrained PF, which is continuous [35].

In addition, the feasible regions of NCTP-1~NCTP-3, NCTP-7~NCTP-9, and NCTP-13~NCTP-15 are very small, while that of the remaining problems f are relatively large.

5.2. Parameter Settings

All algorithms run 30 times independently. The maximum number of iterations for each test function is 500 and the population sizes are set to be 100; and the dimensions of decision vector are set to 30. Additionally, the parameter settings of other comparison algorithms are consistent with the original text.

5.3. Compared with Other Five CMOEAs

To testify the performance of the proposed algorithm, five CMOEAs and 18 CMOPs are used in experiments. Moreover, three performance indicators (i.e., IGD, HV, and FR) are utilized to assess the performances of all compared algorithms. Moreover, the best results are in bold.

IGD: Only the feasible solutions in the population can be used to compute IGD. For all compared algorithms, the mean and standard deviation values of IGD are shown in Table 2. The Wilcoxon statistical results are also given in Table 2. As shown in Table 2, the ACHT-CMODE is superior to SP-CMODE, CDP-CMODE and ATM-CMODE by 10, 10, and 11 test instances, respectively. Moreover, these three CMODEs cannot outperform the proposed algorithm on any functions. It can be concluded that the ACHT can assist the CMODE in selecting a suitable CHT to solve different types of CMOPs. Also, Table 2 indicates that the TOP-NSGAII-CDP and the HypE-FR cannot surpass the ACHT-CMODE on any test functions. Additionally, the performance rankings of the IGD mean values of all compared algorithms achieved by the Friedman statistical analysis are shown in Figure. 1. Clearly, the overall performance of the proposed algorithm is the best among all compared algorithms. In other words, the ACHT-CMODE can achieve a good PF approximation.

Based on the above comparisons, the ACHT-CMODE can outperform other five compared algorithms on most of functions. The main reason is that the proposed ACHT is capable of choosing an appropriate CHT to solve particular types of CMOPs. Therefore, the proposed method is an effective way to solve different types of CMOPs.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

Table 2. Results of all compared algorithm in terms of *IGD*

	SP-CMODE	CDP-CMODE	ATM-CMODE	TOP-NSGAII-CDP	HypE-FR	ACHT- CMODE
NCTP-1	5.13E-02+ (1.01E-01)	6.18E-02+ (9.06E-02)	4.99E-02+ (1.05E-01)	\ \	\ \	**1.51E-02 (4.18E-03)**
NCTP-2	6.54E-02+ (8.82E-02)	9.17E-02+ (1.06E-01)	5.99E-02+ (7.97E-02)	\ \	\ \	**2.71E-02 (1.73E-03)**
NCTP-3	3.21E-02≈ (3.94E-02)	7.92E-02+ (1.27E-01)	3.32E-02≈ (2.45E-02)	\ \	\ \	**2.52E-02 (2.10E-02)**
NCTP-4	8.91E-01+ (2.01E+00)	1.77E-01≈ (7.88E-01)	6.71E-01+ (1.47E+00)	2.93E+00+ (1.36E+00)	8.31E-01+ (2.28E+00)	**2.62E-02 (2.06E-02)**
NCTP-5	**3.86E-02≈ (1.63E-02)**	4.70E-02≈ (7.84E-02)	3.80E-01+ (1.76E-01)	2.67E+00+ (4.05E-01)	1.03E+00+ (2.71E+00)	5.21E-02 (7.87E-02)
NCTP-6	3.87E-01+ (2.46E-01)	3.50E-01+ (1.84E-01)	4.29E-01+ (4.62E-01)	2.72E+00+ (2.08E-01)	1.33E+00+ (3.01E+00)	**2.68E-02 (2.49E-02)**
NCTP-7	1.19E-02≈ (1.70E-02)	6.62E-03≈ (6.59E-03)	1.25E-02- (1.13E-02)	\ \	\ \	**4.86E-03 (3.42E-03)**
NCTP-8	1.45E-02≈ (1.47E-02)	1.63E-02+ (2.39E-02)	1.65E-02≈ (2.48E-02)	\ \	\ \	**6.14E-03 (3.60E-03)**
NCTP-9	**1.28E-02≈ (1.19E-02)**	1.52E-02≈ (1.05E-02)	1.40E-02≈ (1.00E-02)	\ \	\ \	1.52E-02 (1.77E-02)
NCTP-10	4.01E-02+ (3.64E-02)	4.61E-02+ (3.79E-02)	4.26E-02+ (4.02E-02)	4.46E-01+ (4.86E-02)	2.06E-01+ (4.54E-01)	**8.88E-03 (9.72E-03)**
NCTP-11	8.95E-02+ (7.31E-02)	1.64E-02≈ (1.28E-02)	8.56E-02+ (6.50E-02)	5.55E-01- (1.32E-01)	1.89E-01+ (4.79E-01)	**1.60E-02 (1.14E-02)**
NCTP-12	6.45E-02+ (4.63E-02)	1.35E-01+ (2.91E-01)	8.83E-02+ (3.84E-02)	5.83E-01+ (2.56E-01)	1.57E-01+ (4.36E-01)	**1.16E-02 (1.04E-02)**
NCTP-13	1.38E-02≈ (1.45E-02)	2.10E-02+ (2.18E-02)	1.86E-02+ (1.45E-02)	\ \	\ \	**1.09E-02 (1.22E-02)**
NCTP-14	2.09E-02≈ (2.11E-02)	2.58E-02+ (2.25E-02)	1.86E-02≈ (1.57E-02)	\ \	\ \	**1.28E-02 (1.48E-02)**
NCTP-15	1.53E-02≈ (1.36E-02)	1.61E-02≈ (1.65E-02)	1.81E-02≈ (1.90E-02)	\ \	\ \	**1.02E-02 (9.03E-03)**
NCTP-16	1.48E-02+ (4.85E-02)	6.61E-02≈ (2.71E-01)	6.68E-02≈ (2.86E-01)	5.08E-01+ (5.06E-02)	1.89E-01+ (4.56E-01)	**1.32E-02 (1.07E-02)**
NCTP-17	6.78E-02+ (2.68E-01)	7.68E-02+ (2.71E-01)	7.54E-02+ (3.03E-01)	5.09E-01+ (4.30E-02)	1.96E-01+ (4.68E-01)	**4.71E-03 (2.86E-03)**
NCTP-18	8.06E-02+ (2.80E-01)	1.15E-01≈ (3.73E-01)	2.18E-02≈ (2.75E-02)	5.03E-01+ (4.50E-02)	1.86E-01+ (4.65E-01)	**6.23E-03 (4.69E-03)**
+	10	10	11	18	18	
-	0	0	0	0	0	
≈	8	8	7	0	0	

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

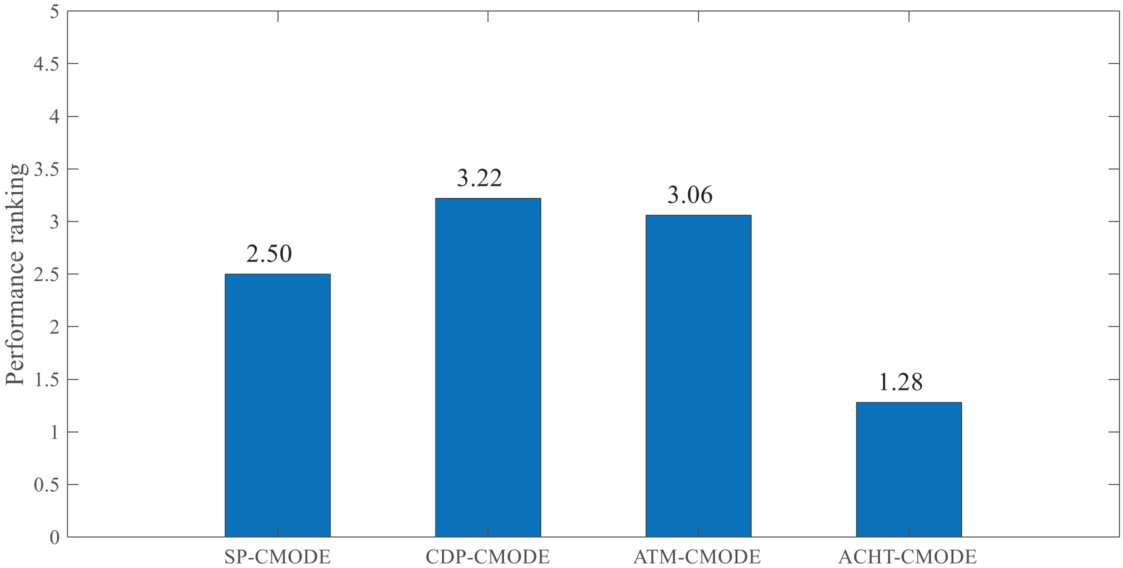


Figure 1. Performance rankings of all compared algorithms in terms of IGD

HV: The reference point is set to [5,5] and only the feasible solutions in the population can be used to compute this indicator in the experiment. The mean and standard deviation values of HV are reported in Table 3. Obviously, the ACHT-CMODE outperforms SP-CMODE, CDP-CMODE and ATM-CMODE on 10, 11, and 10 test instances, respectively, while loses on 1, 1, and 0 instances, respectively. Moreover, TOP-NSGAII-CDP and HypE-FR cannot outperform the proposed algorithm on any functions, and it should be noted that the HV value of TOP-NSGAII-CDP on some test functions is 0, this is because the solution set obtained by TOP-NSGAII-CDP is far from the real PF. Performance rankings of all compared algorithms in terms of HV are depicted in Figure. 2. It can be seen from Figure. 2 that the overall performance of the ACHT-CMODE is the best.

In summary, the convergence and diversity of the PF approximation obtained by the ACHT-CMOD are the best among all compared algorithms. The main reason is that the ACHT-CMODE uses Q-learning algorithm to select suitable CHTs during the entire evolutionary process.

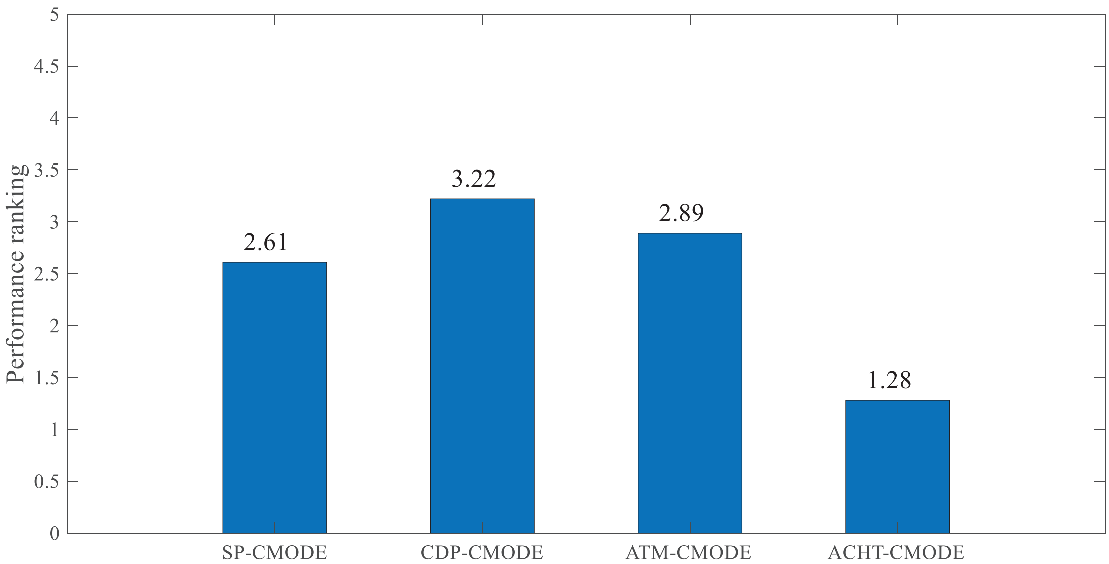


Figure 2. Performance rankings of all compared algorithms in terms of HV

Table 3. Results of all compared algorithm in terms of HV

	SP-CMODE	CDP-CMODE	ATM-CMODE	TOP-NSGAII-CDP	HypE-FR	ACHT- CMODE
NCTP-1	2.72E+01+	2.54E+01+	2.69E+01+	\	\	**2.83E+01**
	(3.12E+00)	(6.73E+00)	(4.23E+00)	\	\	**(9.27E-02)**
NCTP-2	2.57E+01+	2.39E+01+	2.69E+01+	\	\	**2.77E+01**
	(5.00E+00)	(6.54E+00)	(1.53E+00)	\	\	**(7.45E-02)**
NCTP-3	2.78E+01≈	2.57E+01+	2.81E+01≈	\	\	**2.84E+01**
	(2.72E+00)	(5.78E+00)	(7.28E-01)	\	\	**(5.90E-01)**
NCTP-4	1.99E+01+	2.69E+01≈	1.96E+01+	0.00E+00+	2.05E+01+	**2.81E+01**
	(6.37E+00)	(5.13E+00)	(4.57E+00)	(0.00E+00)	(1.07E+01)	**(5.69E-01)**
NCTP-5	2.74E+01≈	**2.73E+01-**	2.05E+01+	0.00E+00+	2.11E+01+	2.71E+01
	(4.58E-01)	**(1.55E+00)**	(3.63E+00)	(0.00E+00)	(8.96E+00)	(1.57E+00)
NCTP-6	2.09E+01+	2.10E+01+	2.10E+01+	0.00E+00+	2.01E+01+	**2.83E+01**
	(5.29E+00)	(4.79E+00)	(5.39E+00)	(0.00E+00)	(9.43E+00)	**(7.74E-01)**
NCTP-7	2.34E+01≈	2.53E+01≈	2.39E+01≈	\	\	**2.58E+01**
	(5.70E+00)	(1.66E+00)	(2.84E+00)	\	\	**(7.29E-01)**
NCTP-8	2.39E+01+	2.32E+01+	2.40E+01≈	\	\	**2.57E+01**
	(2.53E+00)	(3.86E+00)	(3.48E+00)	\	\	**(5.08E-01)**
NCTP-9	**2.43E+01≈**	2.25E+01+	2.41E+01≈	\	\	2.39E+01
	(2.68E+00)	(4.92E+00)	(2.21E+00)	\	\	(3.76E+00)
NCTP-10	2.16E+01+	2.13E+01+	2.18E+01+	0.00E+00+	1.66E+01+	**2.48E+01**
	(4.06E+00)	(3.58E+00)	(4.70E+00)	(0.00E+00)	(8.49E+00)	**(2.43E+00)**
NCTP-11	1.92E+01+	2.39E+01≈	1.84E+01+	0.00E+00+	1.67E+01+	**2.40E+01**
	(5.39E+00)	(2.26E+00)	(6.13E+00)	(0.00E+00)	(9.31E+00)	**(2.04E+00)**
NCTP-12	1.92E+01+	1.75E+01+	1.75E+01+	0.00E+00+	1.70E+01+	**2.47E+01**
	(4.47E+00)	(5.76E+00)	(4.45E+00)	(0.00E+00)	(7.52E+00)	**(2.19E+00)**
NCTP-13	1.91E+01≈	1.80E+01+	1.83E+01+	\	\	**1.99E+01**
	(3.59E+00)	(4.25E+00)	(3.02E+00)	\	\	**(2.65E+00)**
NCTP-14	1.60E+01≈	1.51E+01+	1.63E+01≈	\	\	**1.73E+01**
	(3.66E+00)	(3.76E+00)	(2.80E+00)	\	\	**(2.70E+00)**
NCTP-15	8.99E+00≈	8.94E+00≈	8.56E+00≈	\	\	**9.99E+00**
	(2.81E+00)	(3.09E+00)	(3.60E+00)	\	\	**(2.07E+00)**
NCTP-16	**2.03E+01-**	1.88E+01≈	1.92E+01≈	0.00E+00+	1.41E+01+	1.92E+01
	(4.16E+00)	(4.53E+00)	(4.44E+00)	(0.00E+00)	(7.27E+00)	(2.57E+00)
NCTP-17	1.60E+01+	1.54E+01+	1.57E+01+	0.00E+00+	1.08E+01+	**1.88E+01**
	(4.19E+00)	(5.05E+00)	(4.73E+00)	(0.00E+00)	(6.36E+00)	**(7.21E-01)**
NCTP-18	7.80E+00+	9.24E+00≈	8.89E+00≈	0.00E+00+	4.88E+00+	**1.09E+01**
	(4.37E+00)	(4.28E+00)	(3.68E+00)	(0.00E+00)	(3.50E+00)	**(1.15E+00)**
+	10	11	10	18	18	
-	1	1	0	0	0	
≈	7	6	8	0	0	

FR: The FR values of all CMOEAs on 18 test functions are shown in Table 4. It can be observed from Table 4 that the ACHT-CMODE can find feasible solutions on all CMOPs with 100% probability except for NCTP-14. The ACHT-CMODE is respectively superior to SP-CMODE, CDP-CMODE and ATM-CMODE on 7, 7, and 5 test functions, respectively, and inferior to the above

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

compared algorithms by 1, 0, 0 test functions, respectively. It is also interesting to observe from Table 4 that TOP-NSGAII-CDP and HypE-FR can't find any feasible solutions (i.e., FR values are equal to 0) on the instances with small feasible regions. Additionally, the performance rankings of all compared algorithms obtained by the Friedman's test are illustrated in Figure. 3. As shown in Figure. 3, the overall performance of the ACHT-CMODE is the best in terms of FR.

To sum up, the ACHT-CMODE has the highest probability of finding feasible solutions among all compared algorithms. The main reason is that ACHT can automatically select suitable CHTs when the proposed algorithm solves different types of CMOPs at different stages of the evolution.

Table 4. Results of all compared algorithm in terms of *FR*

	SP-CMODE	CDP-CMODE	ATM-CMODE	TOP-NSGAII-CDP	HypE-FR	ACHT- CMODE
NCTP-1	93.33%	83.33%	100.00%	0.00%	0.00%	100.00%
NCTP-2	96.67%	93.33%	93.33%	0.00%	0.00%	100.00%
NCTP-3	100.00%	100.00%	100.00%	0.00%	0.00%	100.00%
NCTP-4	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-5	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-6	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-7	96.67%	86.67%	100.00%	0.00%	0.00%	100.00%
NCTP-8	90.00%	96.67%	100.00%	0.00%	0.00%	100.00%
NCTP-9	96.67%	90.00%	93.33%	0.00%	0.00%	100.00%
NCTP-10	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-11	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-12	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-13	96.67%	90.00%	96.67%	0.00%	0.00%	100.00%
NCTP-14	100.00%	86.67%	93.33%	0.00%	0.00%	96.67%
NCTP-15	96.67%	100.00%	93.33%	0.00%	0.00%	100.00%
NCTP-16	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-17	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
NCTP-18	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

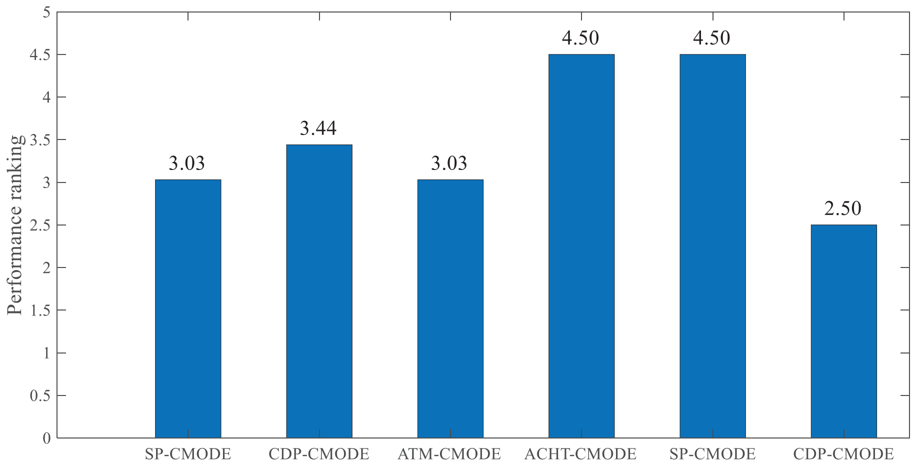


Figure 3. Performance rankings of all compared algorithms in terms of *FR*

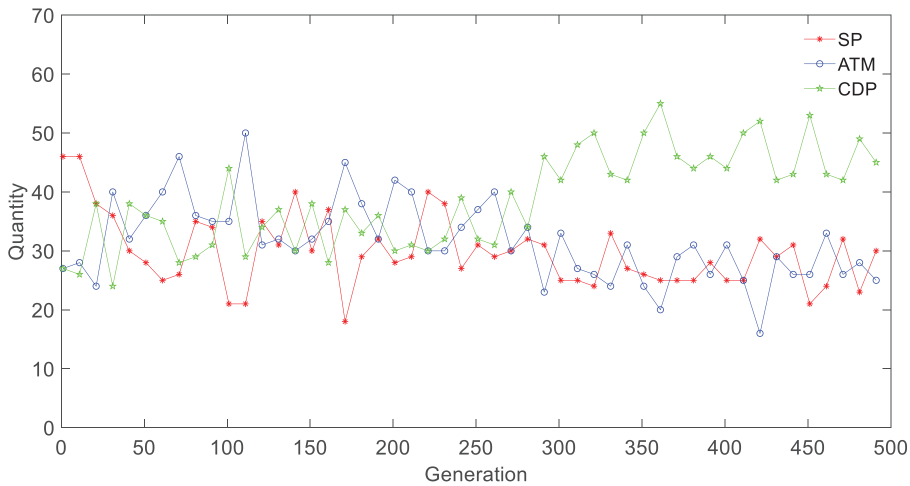
5.4. Experimental Analyses

1) The effectiveness of the ACHT

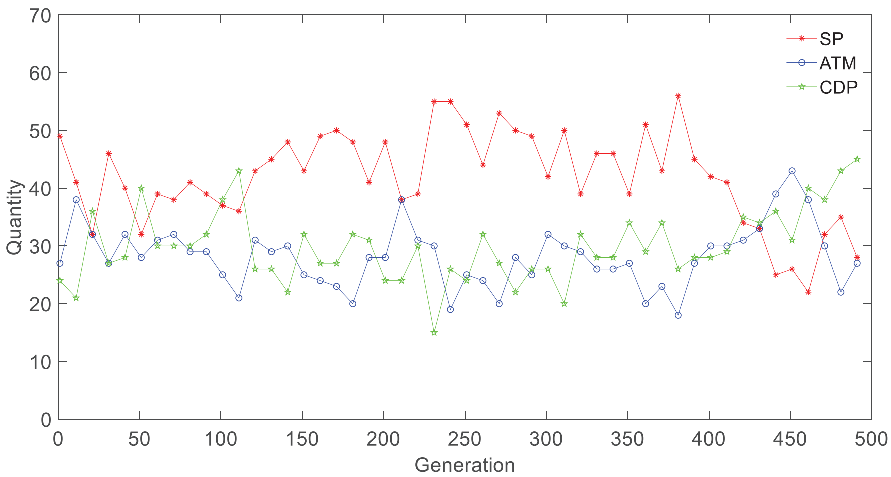
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

To further demonstrate the effectiveness of the proposed method, NCTP-11 and NCTP-12 are used in this experiment. The evolution curves of CHTs are illustrated in Figure. 4.

Figure.4 shows the number of individuals for each CHT on NCTP-11 and NCTP-12 during the entire evolutionary process. It can be seen from Figure. 4 (a) that the performance of three CHTs are similar in the early stage of the evolution, while the CDP plays an important role in the later stages. Therefore, the conclusion conforms to the results shown in Tables 2 and 3 on NCTP-11. It can be observed from Figure. 4 (b) that the effectiveness of the CDP and the ATM are the similar on function NCTP-12. This is consistent with the results shown in Table 3. Moreover, as shown in Figure. 4 (b), the ACHT mainly select the SP to solve the NCTP-12. From Table 3, we can observe that the SP-CMODE outperforms the CDP-CMODE and the ATM-CMODE on this function. Therefore, it can be concluded that the ACHT is an effective method to automatically select a suitable CHT for solving different types of CMOPs.



(a) NCTP-11



(b) NCTP-12

Figure 4. The number changes of each CHT of ACHT-CMODE.

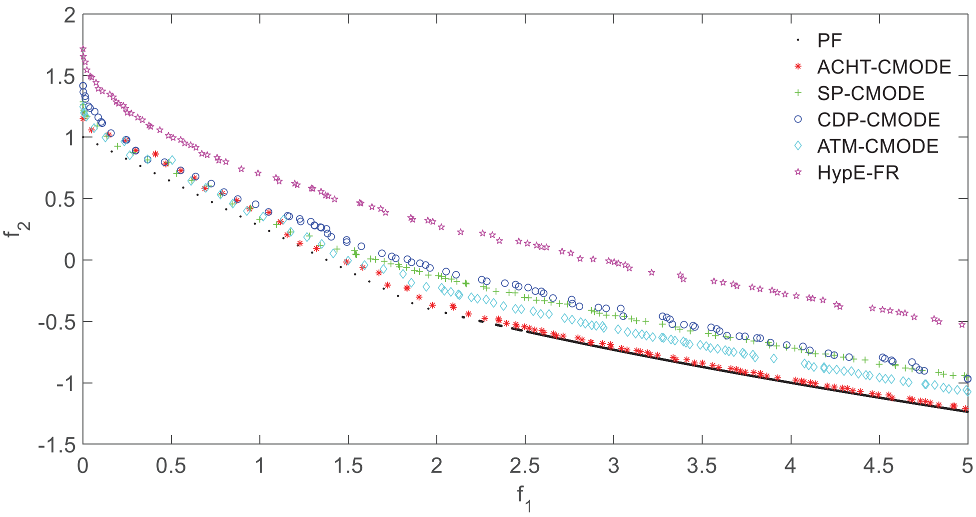
2) Visual comparison on PF approximation

To testify the performance of the proposed algorithm in the objective space, two typical PF approximations of five compared algorithms on two functions (i.e., NCTP-10, and NCTP-18) are plotted in Figure. 5. Note that the solution set obtained by the TOP-NSGAII-CDP is far from the real PF, thus its PF approximations are not given in this subsection.

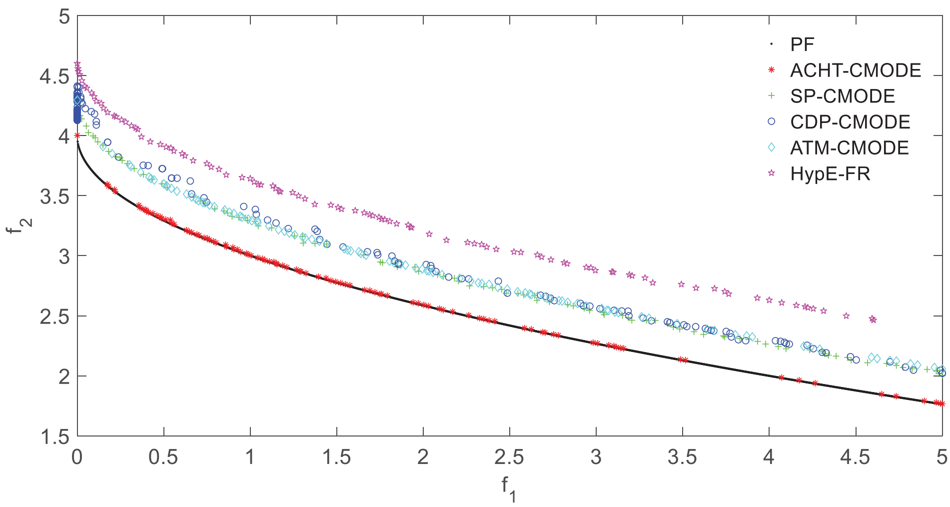
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

As shown in Figure. 5, the whole performance of the PF approximation obtained by the ACHT-CMODE is the best among all algorithms, especially in terms of proximity. But it can be seen from Figure. 5 (b) that its distribution is slightly insufficient. The main reason is that the diversity of the PF approximation obtained by ATM-CMODE is also not good. In detail, the performance of ATM-CMODE is better than that of CDP-CMODE and ATM-CMODE as shown in Table 2, so the proposed algorithm tends to select ATM to handle constraints.

In conclusion, the ACHT can assist the CMODE algorithm in realizing adaptive selection of constraint handling technology, thus it is an effective method to solve CMOPs.



(a) NCTP-10



(b) NCTP-18

Figure 5. Images of the feasible solutions provided by CMOEAs in a run

3) Parametric analyses

The probability ε in the ε-greedy method is mainly used to balance exploration and exploitation searches [11]. Therefore, the performance of the proposed algorithm is impacted by the value of ε.

In this experiment, the sensitive of the parameter ε in the ε-greedy method is analyzed. Above 18 test functions are used to test different ε. Figure. 6 shows the performance rankings of IGD under different ε values. It can be seen from Figure. 6 that the overall performance of the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

algorithm is the best when ε = 0.2. Therefore, the value of ε is set to be 0.2 in the proposed algorithm.

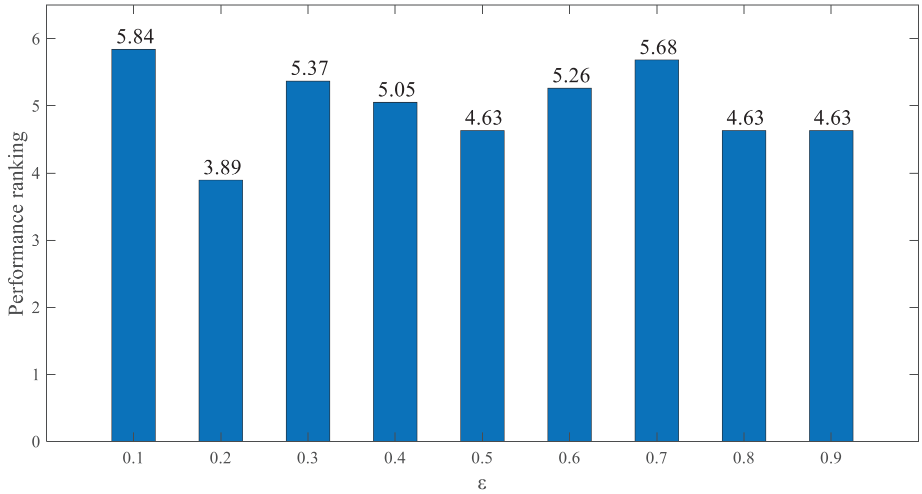


Figure 6. Performance rankings of IGD under different ε values

6. CONCLUSION

To alleviate the selection difficulty problem of constraint handling methods, a constrained multi-objective differential evolution algorithm with adaptive constraint handling technique (ACHT-CMODE) is developed in this study. In the proposed algorithm, three popular constraint handling techniques are selected in the pool and can be self-evolved via a Q-learning method. The performance of the ACHT-CMODE is compared with five other CMOEAs on 18 CMOPs. The results indicate that the proposed method is capable of assist the CMODE in finding high-quality solutions. Moreover, the analysis of the choice of constraint handling technology further verifies the effectiveness of ACHT.

REFERENCES

[1] R. Char, A. Savvaris, A. Tsourdos, et al. Solving multiobjective constrained trajectory optimization problem by an extended evolutionary algorithm [J]. IEEE Transactions on Cybernetics, Vol.50 (2020) No.4, p.1630-1643.

[2] A. Maminov, M. Posypkin. Constrained multi-objective robot's design optimization. 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering. (St. Petersburg and Moscow, Russia, 2020), p.1992-1995.

[3] N.G. Paterakis, M. Gibescu, A.G. Bakirtzis, et al. A multi-objective optimization approach to risk-constrained energy and reserve procurement using demand response [J]. IEEE Transactions on Power Systems, Vol.33 (2018) No.4, p.940-3954.

[4] Z.Z. Liu, Y. Wang. Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces [J]. IEEE Transactions on Evolutionary Computation, Vol.23 (2019) No.5, p.870-884.

[5] M.A. Jan, Q.F. Zhang. MOEA/D for constrained multiobjective optimization: Some preliminary experimental results, 2010 UK Workshop on Computational Intelligence (UKCI), (2010), p.1-6.

[6] YW.Q. Ying, W.P. He, Y.X. Huang, et al. An adaptive stochastic ranking mechanism in MOEA/D for constrained multi-objective optimization. 2016 International Conference on Information System and Artificial Intelligence (ISAI), (Hong Kong, 2016), p.514-518.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

[7] R. Datta, K. Deb and A. Segev. A bi-objective hybrid constrained optimization (HyCon) method using a multi-objective and penalty function approach. 2017 IEEE Congress on Evolutionary Computation (CEC), (San Sebastian, 2017), p.317-324.

[8] Z. Fan, W. Li, X. Cai, et al. Push and pull search for solving constrained multi-objective optimization problems [J]. Swarm Evolutionary Computation, Vol.44 (2019), p.665–679

[9] Q.Q. Fan and X.F. Yan. Self-adaptive differential evolution algorithm with zoning evolution of control parameters and adaptive mutation strategies [J]. IEEE Transactions on Cybernetics, Vol.46 (2016) No.1, p.219-232.

[10] Q.Q. Fan, W.L. Wang, X.F. Yan. Multi-objective differential evolution with performance-metric-based self-adaptive mutation operator for chemical and biochemical dynamic optimization problems [J]. Applied Soft Computing, Vol.59 (2017), p.33-44.

[11] Z.H. Zhou. Machine Learning (Beijing: Tsinghua University Press, China 2016), p.372-388.

[12] X.Y. Cai, Z. Yang, Z. Fan, et al. Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization [J]. IEEE Transactions on Cybernetics, Vol.47 (2017) No.9, p.2824-2837.

[13] Q.F. Zhang, W. Liu, H. Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. 2009 IEEE Congress on Evolutionary Computation, (Trondheim, 2009), p.203-208.

[14] J.P. Li, Y. Wang, S.X. Yang, et al. A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization. 2016 IEEE Congress on Evolutionary Computation (CEC), (Vancouver, BC, 2016), p.4175-4182.

[15] Y.G. Woldesenbet, G.G. Yen, B.G. Tessema. Constraint handling in multiobjective evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation, Vol.13 (2009) No.3, p.514-525.

[16] K. Deb, A. Pratap, S. Agarwal, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. IEEE Transactions on Evolutionary Computation, Vol.6 (2002) No.2, p.182-197.

[17] Y. Wang, Z.X. Cai, Y.R. Zhou, et al. An adaptive tradeoff model for constrained evolutionary optimization. IEEE Transactions on Evolutionary Computation, Vol.12 (2008) No.1, p.80-92.

[18] P.A.N. Bosman, D. Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. IEEE Transactions on Evolutionary Computation, Vol.7 (2003) No.2, p.174-188.

[19] E. Zitzle, L. Thiele, A.E. Eiben, et al. Multiobjective optimization using evolutionary algorithms—a comparative case study. In Proc. Int. Conf. Parallel Problem Solving from Nature, (Berlin, Heidelberg, 1998), p.292–301.

[20] Jadaan O A, Rao C R, L. Rajamani. Solving constrained multi-objective optimization problems using non-dominated ranked genetic algorithm, 2009 Third Asia International Conference on Modelling & Simulation, (Bali, 2009), p.113-118.

[21] B.Y. Qu and P.N. Suganthan. Constrained multi-objective optimization algorithm with diversity enhanced differential evolution, IEEE Congress on Evolutionary Computation (Barcelona, 2010), p.1-5.

[22] M.N. Hsieh, T.C. Chiang and L.C Fu. A hybrid constraint handling mechanism with differential evolution for constrained multiobjective optimization, 2011 IEEE Congress of Evolutionary Computation (CEC), (New Orleans, LA, 2011), p.1785-1792.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202103_7(3).0042

[23] J. Zhang, W. Wang, X. Xu, et al. A multi-objective particle swarm optimization for dual-resource constrained shop scheduling with resource flexibility. 2013 IEEE Symposium on Computational Intelligence for Engineering Solutions (CIES), (Singapore, 2013), p.29-34.

[24] Z.X. Yang, X.Y. Cai and Z. Fan. Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results [J]. Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation. ACM, 2014, p.1181–1186.

[25] X.B. Yu, X.R. Yu, Y.Q. Lu, et al. Differential evolution mutation operators for constrained multi-objective optimization [J]. Applied Soft Computing, Vol.67 (2018), p.452-466.

[26] M.J. Ming, R. Wang, T. Zhang, et al. A dual-grid dual-phase strategy for constrained multi-objective optimization, 2019 IEEE Symposium Series on Computational Intelligence (SSCI), (Xiamen, 2019), p.1881-1888.

[27] H. Zhang, K. Tao, L.B. Ma, et al. Handling constrained multi-objective optimization with objective space mapping to decision space based on extreme learning machine, 2020 IEEE Congress on Evolutionary Computation (CEC), (United Kingdom, 2020), p.1-7.

[28] D.H. Wolpert, W.G. Macready. No free lunch theorems for optimization [J]. IEEE Transactions on Evolutionary Computation, Vol.1 (2017) No.1, p.67–82.

[29] S. Jamal, A. Mohammad, M. Masoud. A reinforcement learning approach to parameter estimation in dynamic job shop scheduling [J]. Computers & Industrial Engineering, Vol.110 (2017), p.75-82.

[30] Y Wang, J.P. Li, X.H. Xue, et al. Utilizing the correlation between constraints and objective function for constrained evolutionary optimization [J]. IEEE Transactions on Evolutionary Computation, Vol.24 (2020) No.1, p.29-43.

[31] R.B. Agrawal, K. Deb. Simulated binary crossover for continuous search space [J]. Complex Systems, Vol.9 (1994) No.3, p.115-148.

[32] Z.Z. Liu, Y. Wang and B.C. Wang. Indicator-based constrained multiobjective evolutionary algorithms [J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2019, p.1–13.

[33] F. Wilcoxon. Individual comparisons by ranking methods [J]. Biometrics bulletin, Vol.1 (1945) No.6, p.80-83.

[34] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance [J]. Journal of the American Statistical Association, Vol.32 (1937) No.200, p.675–701.

[35] Z.W. Ma, and Y. Wang. Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons [J]. IEEE Transactions on Evolutionary Computation, 2019, p.1-1.