Complex & Intelligent Systems
https://doi.org/10.1007/s40747-024-01379-2

ORIGINAL ARTICLE

A novel tri-stage with reward-switching mechanism for constrained multiobjective optimization problems

Jiqing Qu[1] · Xuefeng Li[1,2] · Hui Xiao[1]

Received: 11 November 2023 / Accepted: 3 February 2024

Abstract

The effective exploitation of infeasible solutions plays a crucial role in addressing constrained multiobjective optimization problems (CMOPs). However, existing constrained multiobjective optimization evolutionary algorithms (CMOEAs) encounter challenges in effectively balancing objective optimization and constraint satisfaction, particularly when tackling problems with complex infeasible regions. Subsequent to the prior exploration, this paper proposes a novel tri-stage with reward-switching mechanism framework (TSRSM), including the push, pull, and repush stages. Each stage consists of two coevolutionary populations, namely Pop_1 and Pop_2. Throughout the three stages, Pop_1 is tasked with converging to the constrained Pareto front (CPF). However, Pop_2 is assigned with distinct tasks: (i) converging to the unconstrained Pareto front (UPF) in the push stage; (ii) utilizing constraint relaxation technique to discover the CPF in the pull stage; and (iii) revisiting the search for the UPF through knowledge transfer in the repush stage. Additionally, a novel reward-switching mechanism (RSM) is employed to transition between different stages, considering the extent of changes in the convergence and diversity of populations. Finally, the experimental results on three benchmark test sets and 30 real-world CMOPs demonstrate that TSRSM achieves competitive performance when compared with nine state-of-the-art CMOEAs. The source code is available at https://github.com/Qu-jq/TSRSM.

Keywords Constrained multiobjective optimization · Evolutionary algorithm · Tri-stage · Reward-switching mechanism

Introduction

Many real-world problems belong to constrained multiobjective optimization problems (CMOPs), which have conflicting objectives subject to various constraints [1–4]. The general CMOP can be expressed as follows:

$$\min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x})) \tag{1}$$

s.t.

$$\begin{cases} g_i(\mathbf{x}) \leq 0, i = 1, \ldots, s \\ h_i(\mathbf{x}) = 0, i = s+1, \ldots, t \\ \mathbf{x} \in \Omega, \end{cases} \tag{2}$$

where $\mathbf{F}(\mathbf{x})$ is composed of M conflicting objective functions; $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ is a solution with D dimensions; $\Omega \in \mathbb{R}^n$ denotes the decision space; $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ indicate s inequality constraints and $t - s$ equality constraints, respectively; and t is the number of constraints.

To express the degree of the ith constraint violation (denoted as $CV_i(\mathbf{x})$) of $\mathbf{x}$ at the ith constraint, the following formulation is used:

$$CV_i(\mathbf{x}) = \begin{cases} \max(0, g_i(\mathbf{x})), \ i = 1, \ldots, s \\ \max(0, |h_i(\mathbf{x})| - \delta), \ i = s+1, \ldots, t, \end{cases} \tag{3}$$

where δ is a very small positive constraint boundary relaxation parameter (e.g., 1e–4), which turns $h_i(\mathbf{x})$ into inequality constraints. The overall constraint violation value of $\mathbf{x}$

✉ Hui Xiao
xiaohui@tongji.edu.cn

Jiqing Qu
jqqu@tongji.edu.cn

Xuefeng Li
lixuefeng@tongji.edu.cn

[1] College of Electronic and Information Engineering, Tongji University, 4800 Cao'an Highway, Shanghai 201804, China

[2] Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, 55 Hechuan Road, Shanghai 201804, China

Published online: 30 March 2024

(denoted as CV) is formulated as:

$$CV(\mathbf{x}) = \sum_{i=1}^{t} CV_i(\mathbf{x}), \tag{4}$$

when $CV(\mathbf{x}) = 0$ means that the decision variable $\mathbf{x}$ is a feasible solution. Otherwise, it is an infeasible solution. Given $\mathbf{x}_1$ and $\mathbf{x}_2$ are all feasible, if $\mathbf{F}(\mathbf{x}_1)$ is not worse than $\mathbf{F}(\mathbf{x}_2)$ and it at least has one better objective, $\mathbf{x}_1$ is said to dominate $\mathbf{x}_2$ ($\mathbf{x}_1 \prec \mathbf{x}_2$). A solution is deemed Pareto-optimal if no other feasible solution dominates it. CMOPs aim to find a set of Pareto-optimal solutions that satisfy various constraints. In the decision space, the set of all feasible Pareto-optimal solutions is the Pareto-optimal set (PS). The mapping of the PS onto the objective space forms the constrained Pareto front (CPF). Similarly, when addressing unconstrained multiobjective optimization problems (MOPs), the unconstrained Pareto front (UPF) is ultimately desired [5].

In contrast to unconstrained MOPs, CMOPs pose a greater challenge in simultaneously managing conflicting objectives and constraints [6, 7]. Many constrained multiobjective evolutionary algorithms (CMOEAs) have been developed to address this issue by employing diverse constraint-handling techniques (CHTs). The current CHTs can be divided into five categories: (1) penalty function methods [8, 9]; (2) separation of constraints and objectives [10, 11]; (3) multiobjective methods [12, 13]; (4) hybrid methods [14, 15]; and (5) multi-stages and multi-populations (MSMP) [16, 17]. Although those methods employed in the state-of-the-art CMOEAs have demonstrated high performance on certain CMOPs, they still have limitations when it comes to solving problems with complex infeasible regions and small discrete feasible regions. Unfortunately, many real-world problems exhibit such characteristics, such as the problem of synchronous optimal pulse-width modulation of 3-level inverters [18], which pose challenges to the existing CMOEAs. More specifically, the penalty function methods and separation of constraints and objectives require careful tuning of related parameters. Designing an additional objective becomes challenging in multiobjective methods. Hybrid methods demand differentiability of the problem [19]. MSMP-based CMOEAs overcome the challenges of directly solving CMOPs by leveraging infeasible solutions to extract valuable information, which facilitates the collaboration between populations and stages [20]. However, they encounter difficulties in effectively leveraging infeasible solutions.

Inspired by the success of MSMP, a novel tri-stage with reward-switching mechanism framework (TSRSM) is proposed for CMOPs. The three stages of TSRSM employ distinct strategies to leverage infeasible solutions. The novel features of TSRSM are as follows:

1. The proposed TSRSM framework is comprised of three stages: the push stage, the pull stage, and the repush stage. Each stage employs two cooperative populations, namely Pop_1 and Pop_2. The role of Pop_2 varies across different stages. In the push stage, Pop_2 aims to converge to the UPF and guide Pop_1 to pass through infeasible regions. Subsequently, Pop_2 employs the constraint relaxation technique to enhance feasibility in the pull stage. Finally, Pop_2 reconvenes with the UPF using knowledge transfer and shares its unique insights to inform and guide Pop_1 in the repush stage. The novel characteristic of this approach is that the Pop_2 alternates between the UPF and CPF, resulting in greater effectiveness compared to the single-direction movement of the auxiliary population in existing CMOEAs (e.g., CCMO), as evidenced by experimental results.
2. A novel reward-switching mechanism (RSM) is devised to decide when to switch stages by evaluating the convergence and diversity levels exhibited by the population. One distinct characteristic of this approach is that RSM takes into account the convergence and diversity of the population simultaneously, making it a more accurate method to switch stages compared to other switching mechanisms.

To demonstrate the performance of TSRSM, 9 state-of-the-art CMOEAs were selected for comparison on three benchmark test sets and 30 real-world CMOPs [21]. The results reveal that the proposed method achieves superiority over other CMOEAs on both benchmark problems and real-world CMOPs. Additionally, TSRSM obtains the best performance on 10 real-world problems, including the synchronous optimal pulse-width modulation of 3-level inverters problem, the multi-product batch plant problem [22], the heat exchanger network design problem [23], and others. This achievement represents the highest number of best results compared to other CMOPs.

It should be noted that while many research works are based on multi-stage approaches [24, 25], they primarily focus on utilizing different tasks in each stage, rather than emphasizing the optimization problems themselves. In the TSRSM, the current stage can continue to evolve based on its performance in the problem at hand. This means that if the pull stage performs well in the problem, there may be no need for a repush stage.

The remainder of this paper is organized as follows. In "Related works and motivation", we review the existing MSMP and explain our motivations. In "Proposed method", the proposed TSRSM is introduced. "Experimental results" shows the experimental setting and results. Finally, "Conclusions and future work" presents the conclusions and future work.

Related works and motivation

This section provides an overview of the existing CMOEAs that are relevant to the field of MSMP, as this paper specifically emphasizes the multi-stage framework.

CMOEAs based on two stages and two populations

This part of the method tries to balance objectives and constraints by two stages and two populations.

As for two-population algorithms, one representative is CCMO [26], which is a coevolutionary framework featuring two weak cooperative populations. One population is exclusively dedicated to solving the original CMOPs with the specific objective of finding the CPF. In contrast, the other population focuses its efforts on discovering the UPF. Another two-population algorithm called cDPEA [27], in which one population is designed to preserve competitive infeasible solutions, and the other population adopts a feasibility-oriented approach to handle infeasible solutions. Furthermore, a novel adaptive fitness function was implemented to regulate the trade-off between convergence and diversity.

As for two-stage algorithms, one representative is PPS-MOEA/D [28], which introduced a push–pull searching strategy. The push stage is mainly focused on directing the population toward the UPF, while the pull stage is responsible for attracting the population toward the CPF. The switching mechanism employs the gradient of the maximum of the nadir and the minimum of ideal points. DD-CMOEA [5] employed this switching mechanism in the exploration and exploitation stage with two populations. The primary objective of the exploration stage is to search for informative infeasible solutions. In contrast, the exploitation stage leverages infeasible solutions to explore nearby feasible solutions. CMOEA-MS [29] is another two-stage algorithm that consists of a first stage for identifying feasible regions and a second stage for spreading along feasible boundaries. Moreover, CMOEA-MS used fitness evaluation strategies to adaptively balance objectives and constraints in two stages. Another strategy was developed by TSTI [30], which employed different emphases on the three indicators (namely convergence, diversity, and feasibility) in two stages. The first stage is to obtain solutions with good distribution and to prevent the population from falling into local optima. The second stage is to quickly converge to the CPF. DATEA [31] used weak coevolution of the dual population to consider constraints in the first stage. Then, a feasibility-oriented approach is employed to guide a single population in spreading across the feasible regions discovered in the first stage. URCMO [32] utilized the knowledge learned from the learning stage about the relationship between the UPF and the CPF to guide the evolving strategies in the evolving stage.

Inspired by the success of evolutionary multitask (EMT) in other fields, such as high-dimensional classification feature selection problems, some researchers have attempted to develop EMT to solve CMOPs. Qiao et al. [33] first introduced EMT [34] into CMOEAs (named EMCMO), which includes two tasks: the first task is designed to solve the original CMOP, and the other is for the unconstrained MOP. Furthermore, a transfer strategy was devised to determine whether to transfer parent or offspring sets into the environmental selection. A novel EMT, named MTCMO [35], was subsequently developed, which employs a dynamic auxiliary task and leverages an improved ϵ-constraint method to effectively tackle complex CMOPs. Furthermore, a tri-task framework known as CMOEMT [36] was introduced. Three tasks are designed for the original CMOP, the unconstrained MOP, and the relaxed CMOP, respectively. The evolutionary process can be broken down into two distinct stages: the evolving stage and the transfer stage. During the evolving stage, three specific tasks evolve independently. Conversely, the transfer stage effectively transmits relevant information among the three tasks.

CMOEAs based on three stages and three populations

This part of the method attempts to balance objectives and constraints in a more granular way.

TriP [37] is a representative three-population algorithm. Two populations evolve using a weak coevolutionary framework to handle the original CMOP and the unconstrained MOP separately, while the third population independently addresses the relaxed CMOP. Three populations of TriP and CMOEMT have the same purpose, but the way they exchange information with each other is different. C3M [24] is one representative three-stage algorithm. In the early stage, setting aside the typical consideration of feasibility to enable a more thorough exploration of the objective space. At the medium stage, the algorithm focuses on individual constraints, selecting those of the highest priority to explore the objective space further. In the last stage of the algorithm, feasibility is fully accounted for to enhance the quality of solutions achieved in the previous two stages. Another three-stage algorithm, TSCSO [25], introduced a tri-stage competitive swarm optimizer. The first stage focuses on achieving global convergence to the UPF, the second stage aims to enhance the diversity of the population and explore more feasible regions, and the third stage is utilized to search for the feasible regions omitted in the previous stage.

Motivations

The aforementioned studies share a common objective of addressing CMOPs by utilizing distinct populations or stages to handle the CMOP, unconstrained MOP, and relaxed CMOP. This is because MSMP-based evolutionary algorithms can circumvent the challenges encountered in directly solving the original CMOPs by employing a well-designed staged approach [19]. First, the stage of solving the unconstrained MOP helps to find promising solutions by ignoring constraints. Second, the stage of solving the relaxed CMOP is beneficial to expanding feasible solutions. Third, the stage of solving the original CMOP allows the population to converge further to the CPF. However, the inadequate weight they possess in the evolutionary process can lead to challenges in solving specific problems. For example, if the algorithm neglects the utilization of the relaxed CMOP, it struggles to solve problems characterized by large infeasible regions and small discrete feasible regions. Hence, designing an effective framework and switching mechanism is crucial for achieving optimal results.

The existing frameworks still have some weaknesses. CCMO utilized a coevolutionary framework with two weak cooperative populations, which search for the CPF and UPF, respectively. However, searching for the UPF in the later stages results in significant resource wastage, which means that it may be ineffective to search for the CPF when the UPF and CPF are located far apart. TriP used the tri-population-based coevolutionary framework, which solves the CMOP, the unconstrained MOP, and the relaxed CMOP, respectively. The third population uses the ϵ-constrained technique in PPS-MOEA/D, easily falling into the local optimum, such as MW1, MW2, and MW10 [38]. CMOEMT encounters a similar challenge to Trip as it also utilizes the ϵ-constrained technique independently during the initial stage. EMCMO employed an EMT framework with knowledge transfer, which has been demonstrated to achieve high performance on MW [38] problems. MTCMO improved the EMT framework and knowledge transfer, which also has high performance on MW. However, both EMCMO and MTCMO have limitations in dealing with problems that have large infeasible regions, such as LIR-CMOP [39], because they primarily prioritize feasibility.

Switching mechanisms are designed to achieve a balance among the distinct stages that serve different tasks. However, the existing switching mechanisms still encounter difficulties due to their inaccurate judgment of CMOPs with diverse characteristics as illustrated in Table 1. Consequently, the efficiency and versatility of these existing switching mechanisms remain insufficient.

To solve the aforementioned challenges, we propose a novel TSRSM framework, which is described in "Proposed Method".

Proposed method

The procedure of TSRSM

As presented in Fig. 1, the proposed TSRSM is a cooperative coevolutionary framework, including two populations, namely, Pop_1 and Pop_2, and three stages, namely the push stage, the pull stage, and the repush stage. Figure 2 illustrates the distribution of two populations across three stages. The change in position of Pop_2 is the origin of the name given to the tri-stage. RSM is employed in the stage transition. The pseudo-code of TSRSM is shown in Algorithm 1. At the beginning of TSRSM, Pop_1 and Pop_2 are initialized randomly with size N. Then, α is initialized with 20, which is described in detail in "Reward-switching mechanism". RSM is applied to calculate α and β by Algorithm 2, which determines the evolutionary stage. In the push stage, Pop_1 and Pop_2 are evolved with the weak coevolutionary framework. Pop_2 is designed to deal with the unconstrained MOP and aims to quickly converge to the UPF. In the pull stage, Pop_2 evolves independently using MOEA/D-Epsilon, while Pop_1 uses information from Pop_2 to evolve. In the repush stage, Pop_1 and Pop_2 will utilize knowledge transfer to select a parent set or an offspring set from another population in their respective environmental selection, which is different in the push stage.

It should be noted that the CDP method and fast nondominated sorting method are utilized for environmental selection in both populations. Besides, the fitness assign method and the truncation strategy proposed in SPEA2 [40] are used in TSRSM. The above methods are used because they have shown promising performance in CCMO [26], Trip [37], MTCMO [35], etc. Naturally, Pop_2 evolved by MOEA/D-Epsilon in the pull stage does not employ these methods.

```
Algorithm 1: Procedure of TSRSM
Input: Population size $N$, termination condition
Output: $\text{Pop}_1$
Initialize $\text{Pop}_1$ and $\text{Pop}_2$ of size $N$;
$k = 0$;// $k$ is current generation;
$\alpha$=20;
while Termination condition is not satisfied do
    Calculate $\alpha$, $\beta$ using RSM→ Algorithm 2;
    if $k \le \alpha$ then
        The push stage → Algorithm 3;
    else if $k \le \beta$ then
        The pull stage → Algorithm 4;
    else
        the repush stage → Algorithm 5;
    end
    $k = k + 1$;
end
```

Table 1 Switching mechanism of existing algorithm

Algorithm	Switching mechanism	Shortcoming
PPS-MOEA/D, DD-CMOEA, and TriP	The gradient of the maximum of the nadir and the minimum of the ideal points	Overlooking diversity contributions made by other solutions
CMOEMT, TSTI, URCMO, and EMCMO	The fixed number of evaluations	Incapable of making proper adjustments for complex CMOPs, which demand a greater number of evaluations to navigate through infeasible regions
C3M	All solutions are nondominant, the magnitude of change in objective values, and the considered number of constraints	Assigning a high weight to objectives optimization impedes convergence toward the CPF
CMOEA-MS and DTAEA	The ratio of feasible solutions (RFS)	Facing challenges when dealing with problems that have small feasible regions

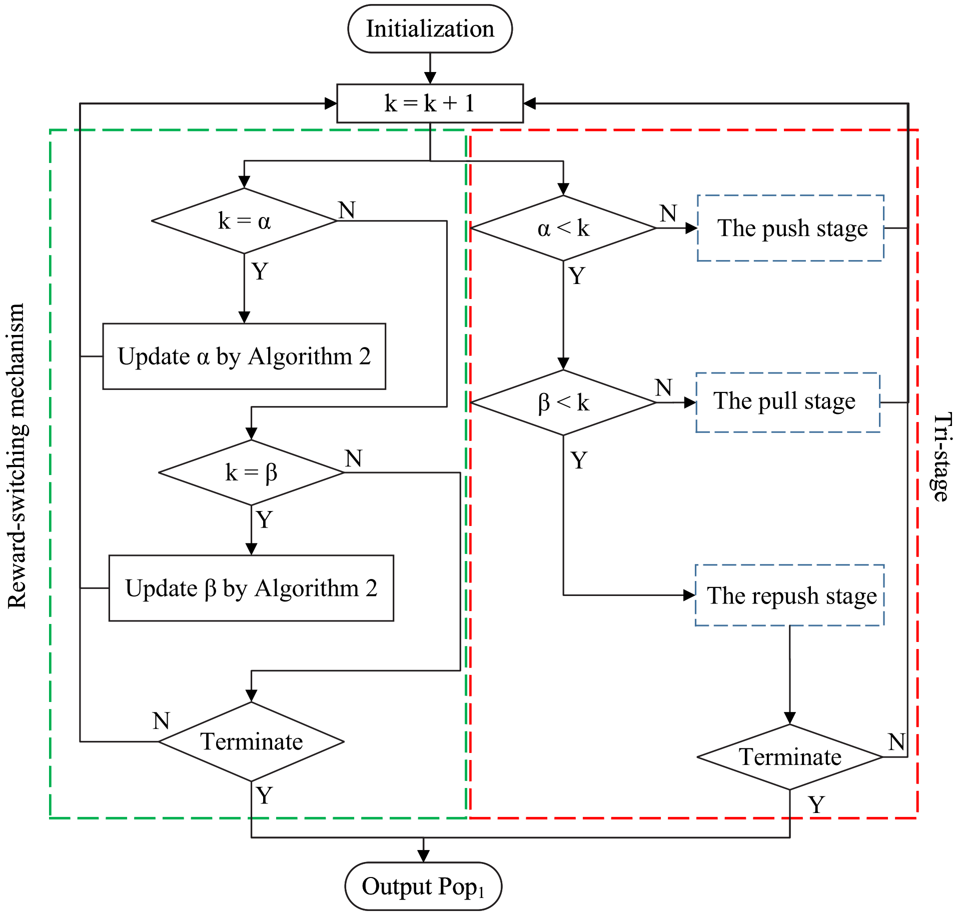


Fig. 1 The flowchart of TSRSM

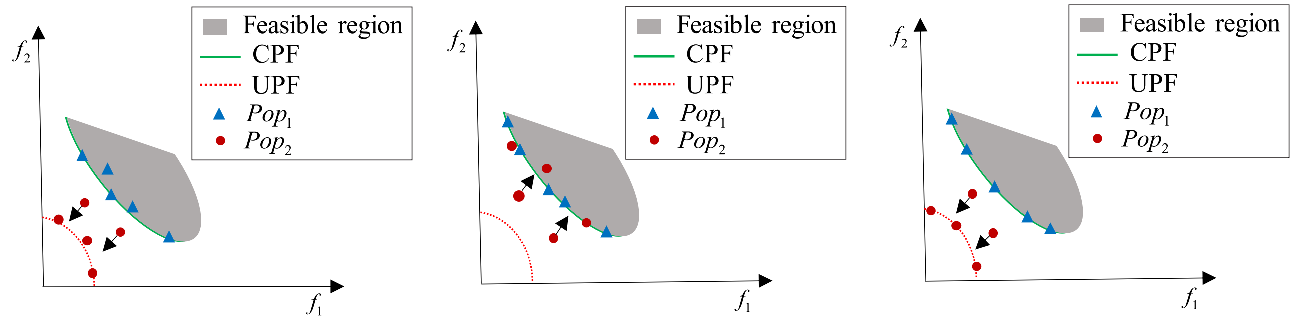
Reward-switching mechanism

As previously discussed, determining the transition between stages is crucial, as the evolutionary process tends to decelerate in later phases, making fixed criteria obsolete. Hence, RSM dynamically adjusts the number of evolutionary generations in the first two stages based on the performance of the respective population.

To assess the performance of the population, RSM uses two indicators: convergence and diversity. The recommended approach for dispensing rewards is as follows:

$$\mathrm{MG}_k = \max(Rc_k, Rd_k) \geq \lambda, \tag{5}$$

where MG_k is the maximum rate of change between the convergence and diversity observed in the last gr generations up to the kth generation, and λ represents the threshold value



(a) During the push stage

(b) During the pull stage

(c) During the repush stage

Fig. 2 The distribution of the two populations in three stages

for the rate of change. If $\mathrm{MG}_k \geq \lambda$, the current stage will be extended for an additional gr generations to facilitate further evolutionary progress. Therefore, MG_k also indicates the performance of the population over the span of gr generations. The rates of change of the convergence and diversity are formulated in Eqs. 6 and 7, respectively.

$$Rc_k = \frac{\left|sc_k - sc_{k-gr}\right|}{\max(sc_{k-gr}, \delta)}, \tag{6}$$

$$Rd_k = \frac{\left|sd_k - sd_{k-gr}\right|}{\max(sd_{k-gr}, \delta)}, \tag{7}$$

$$sc_k = \sum_{i=1}^{N} Ic_i^k, \tag{8}$$

$$sd_k = \sum_{i=1}^{N} Id_i^k, \tag{9}$$

where sc_k and sd_k indicate the sum of Ic_i^k and Id_i^k at the kth generation, respectively, for the corresponding population, as indicated in Eqs. 8 and 9; Ic_i^k and Id_i^k represent the convergence and diversity of the ith individual at the kth generation, respectively; gr is the number of reward generations; N represents population size; and δ is a small parameter (e.g., 1e–6). The calculation of Ic_i^k and Id_i^k is the same as that of TSTI [30]. For reasons of space, this paper does not show here. It should be noted that the calculations of sc_k and sd_k involve the use of Pop_1 in the push stage. The updation of gr is distinct in different stages as follows.

$$gr_i = \begin{cases} 20, & i = 1 \\ 200, & i = 2, \end{cases} \tag{10}$$

where $i = 1, 2$ means in the push stage and pull stage, respectively. To make readers more easily understand, gr means the number of rewarded generations, while the subscript i in the gr indicates the number of generations rewarded in either the push or pull stage. When in the push stage, awarded 20 generations at a time. However, when in the pull stage, 200 generations will be available to reward more generations. This is because, over time, populations face increasing challenges in exploring superior solutions during the evolution process, which is discussed in "Experimental results".

Algorithm 2: Procedure of RSM

```
Input: $\mathrm{Pop}_1$, $\mathrm{Pop}_2$, $\alpha$, $\beta$, Current stage
Output: $\alpha$, $\beta$
if Current stage = push stage and $k = \alpha$ then
    Calculate $\mathrm{MG}_\alpha$ according to Equation 5;
    if $MG_\alpha \geq \lambda$ then
        $\alpha = \alpha + gr_1$;
    else
        $\beta$=$\alpha$+$gr_2$;// $\beta$ is initialized here;
    end
end
if Current stage = push stage and $k = \beta$ then
    Calculate $\mathrm{MG}_\beta$ according to Equation 5;
    if $MG_\beta \geq \lambda$ then
        $\beta = \beta + gr_2$;
    end
end
```

Algorithm 2 gives the procedure of RSM. Firstly, MG_k is calculated by Eq. 5. Then, gr is calculated in different stages. When $\mathrm{MG}_k \geq \lambda$, α and β are updated accordingly. In the push stage, when $\mathrm{MG}_k < \lambda$, β is initialized. It should be noted that MG_k only be calculated when $k = \alpha$ or $k = \beta$ to reduce computational time.

The push stage

The purpose of the push stage is to quickly converge to UPF. The procedure of the push stage is presented in Algorithm 3. Pop_1 and Pop_2 are evaluated by the fitness assign method. Pop_1 considers both objectives and constraints, while Pop_2 only considers objectives. Then, Pop_1 and Pop_2 select their mating pool by binary tournament selection. Simulated binary crossover (SBX) [41] and polynomial mutation (PM) [42] are employed to generate $N/2$ offspring for each popu-

Algorithm 3: Procedure of the push stage

```
Input: $\text{Pop}_1$, $\text{Pop}_2$, $k$
Output: $\text{Pop}_1$, $\text{Pop}_2$
$f_{origin}$ ←Evaluate $\text{Pop}_1$ by original CMOP;
$f_{helper}$ ←Evaluate $\text{Pop}_2$ by unconstrained MOP;
while $k \leq \alpha$ do
    $Parent_1 \leftarrow$ Select $\frac{N}{2}$ individuals from $\text{Pop}_1$ using binary mating selection based on $f_{origin}$ ;
    $Parent_2 \leftarrow$ Select $\frac{N}{2}$ individuals from $\text{Pop}_2$ using binary mating selection based on $f_{helper}$ ;
    $\text{Off}_1 \leftarrow$ Generated by GA operator based on $Parent_1$;
    $\text{Off}_2 \leftarrow$ Generated by GA operator based on $Parent_2$;
    $\text{CP}_1 \leftarrow Pop_1 \cup Off_1 \cup Off_2$;
    $\text{CP}_2 \leftarrow Pop_2 \cup Off_1 \cup Off_2$;
    $f_{origin}$ ←Evaluate $\text{CP}_1$ by original CMOP;
    $f_{helper}$ ←Evaluate $\text{CP}_2$ by unconstrained MOP;
    $\text{Pop}_1 \leftarrow$ Environment selection based on $f_{origin}$;
    $\text{Pop}_2 \leftarrow$ Environment selection based on $f_{helper}$;
end
```

lation. Then, both offspring set is added into Pop_1 and Pop_2, respectively, denoted as CP_1 and CP_2. Next, CP_1 and CP_2 are evaluated by the original CMOP and the unconstrained MOP, respectively. Finally, Pop_1 and Pop_2 are updated based on environment selection.

The pull stage

The purpose of the pull stage is to explore and expand the set of feasible solutions to address CMOPs characterized by complex feasible regions. Algorithm 4 outlines the procedure of the pull stage. Pop_1 evolves in a manner similar to its evolution in the push stage, with the difference that CP_1 consists of the population sets Pop_1, Pop_2, and Off_1. The reason why Off_2 is not used in CP_1 is discussed in "Experimental results". The evolution of Pop_2 is accomplished through the utilization of MOEA/D-Epsilon, following the same approach as the second stage of PPS-MOEA/D [28]. ϵ is updated by Eq. 11.

$$\epsilon(v) = \begin{cases} (1-\gamma)\varepsilon(v-1), & \text{if } rf_v < \varphi \\ \epsilon(0)(1-\frac{v}{T_c})^{sp}, & \text{if } rf_v \geq \varphi, \end{cases} \tag{11}$$

where v represents the vth generation from the beginning of the pull stages; rf_v is the feasible ratio of Pop_2; sp controls the speed at which the relaxation of constraints is reduced; γ and φ are control parameters; $\epsilon(0)$ is the maximal CV of Pop_2 at the end of push stage; and T_c indicates the total number of generations except the push stage. The number of generations in the pull stage is not utilized because it is indeterminate at the beginning. The influence on evolution is minimal, as ϵ are nearly equal in the later phases.

Algorithm 4: Procedure of the pull stage

```
Input: $\text{Pop}_1$, $\text{Pop}_2$, $k$
Output: $\text{Pop}_1$, $\text{Pop}_2$
$f_{origin}$ ←Evaluate $\text{Pop}_1$ by original CMOP;
while $\alpha < k \leq \beta$ do
    $Parent_1 \leftarrow$ Select $\frac{N}{2}$ individuals from $\text{Pop}_1$ using binary mating selection based on $f_{origin}$ ;
    $\text{Off}_1 \leftarrow$ Generated by GA operator based on $Parent_1$;
    $\text{Pop}_2 \leftarrow$ Evolve $\text{Pop}_2$ by MOEA/D-Epslion considering constraints;
    $\text{CP}_1 \leftarrow Pop_1 \cup Pop_2 \cup Off_1$;
    $f_{origin} \leftarrow$ Evaluate $\text{CP}_1$ by original CMOP;
    $\text{Pop}_1 \leftarrow$ Environment selection based on $f_{origin}$;
end
```

The repush stage

Different from the push stage, knowledge transfer [33] is employed in environment selection. The purpose of knowledge transfer is to select a more effective population to attend environment selection. As presented in Algorithm 5, Lines 8–13 are the procedure of knowledge transfer. The effectiveness of the populations relies on the successful transfer rates of parent sets and offspring sets (Rp and Ro), which are calculated as Eqs. 12 and 13:

$$Rp = \frac{\text{num\_}p}{N}, \tag{12}$$

$$Ro = \frac{\text{num\_off}}{\frac{N}{2}}, \tag{13}$$

where $Rp \in [0, 1]$, $Ro \in [0, 1]$, num_p and num_off are, respectively, the number of parent sets and offspring sets in the N best individuals, which are selected from TCP_1 and TCP_2 by environment selection. If $Rp > Ro$, it means parent sets are more effective in accommodating the final environment selection. To maintain diversity, $N/2$ individuals are selected randomly from parent sets for Trans$P$. If $Rp < Ro$, the offspring sets are selected for Trans$P$. Finally, the new populations are obtained by environment selection.

Computational complexity

The proposed TSRSM mainly includes genetic operators, mating selection, and population updation, except for Pop_2 in the pull stage. Since the MOEA/D-Epsilon method uses PPS-MOEA/D in the pull stage, the complexity is $O(M \cdot N^2)$. Moreover, population updation uses SPEA2, so the complexity is $O(M \cdot N^3)$. The complexity of genetic operators and mating selection are, respectively, $O(N \cdot D)$ and $O(N)$. In summary, the worst computational complexity of TSRSM is $O(M \cdot N^3)$. However, the reward-switching mechanism and knowledge transfer method would consume some computational time.

Algorithm 5: Procedure of the repush stage

```
Input: $Pop_1$, $Pop_2$, $k$
Output: $Pop_1$, $Pop_2$
$f_{origin}$ ←Evaluate $Pop_1$ by original CMOP;
$f_{helper}$ ←Evaluate $Pop_2$ by unconstrained MOP;
while $\beta < k$ do
    $Parent_1$ ← Select $\frac{N}{2}$ individuals from $Pop_1$ using binary mating selection based on $f_{origin}$ ;
    $Parent_2$ ← Select $\frac{N}{2}$ individuals from $Pop_2$ using binary mating selection based on $f_{helper}$ ;
    $Off_1$ ← Generated by GA operator based on $Parent_1$;
    $Off_2$ ← Generated by GA operator based on $Parent_2$;
    $TCP_1 \leftarrow Pop_1 \cup Off_1$;
    $TCP_2 \leftarrow Pop_2 \cup Off_2$;
    Evaluate $TCP_1$ on the MOP;
    Evaluate $TCP_2$ on the CMOP;
    $TransP_1$ ← According to Equation 12-13;
    $TransP_2$ ← According to Equation 12-13;
    $CP_1 \leftarrow Pop_1 \cup Off_1 \cup TransP_2$;
    $CP_2 \leftarrow Pop_2 \cup Off_1 \cup TransP_1$;
    $f_{origin}$ ←Evaluate $CP_1$ by original CMOP;
    $f_{helper}$ ←Evaluate $CP_2$ by unconstrained MOP;
    $Pop_1$ ← Environment selection based on $f_{origin}$;
    $Pop_2$ ← Environment selection based on $f_{helper}$;
end
```

Experimental results

To evaluate the performance of the proposed TSRSM in solving CMOPs, a series of experiments are conducted on PlatEMO [43].

Experimental settings

Test functions

To demonstrate the performance of TSRSM, three benchmark test suites (MW [38], LIRCMOP [39], and constraint DTLZ [44, 45]) are adopted. The details of those are in Table 2.

Compared algorithms

To perform the effectiveness of the proposed TSRSM, the 9 most state-of-the-art algorithms are selected as follows:

1. Two-population based: CCMO [26] and cDPEA [27].
2. Two-stage based: CMOEA-MS [29], DSPCDE [16], and TSTI [30].
3. EMT-based: MTCMO [35] and CMOEMT [36].
4. Three-population based: TriP [37].
5. Three-stage based: C3M [24].

In algorithms that use GA as an operator, the parameters of SBX and PM are set as follows:

1. Crossover probability of SBX: 1
2. Mutation probability of PM: $1/D$

In algorithms that use DE as an operator, the parameters CR and F are 1 and 0.5, respectively.

The population size N is set to 91, taking into account the influence of the weight vector set on the actual population size in PlatEMO [46]. This consideration serves to render the experimental results more reliable and consistent [36]. The maximum number of function evaluations $E_{\max}$ is set to 100,000 for all test functions. Moreover, each algorithm runs independently for 30 times on each test function. The parameter settings of all the methods in comparison are the same as suggested in their original literature. TSRSM uses the same parameter for constraint relaxation as the PPS-MOEA/D. Specifically, the parameters λ, gr_1, and gr_2 in TSRSM are set to 1e–2, 20, and 200, respectively.

Performance indicators

To measure the performance of each algorithm, inverted generational distance based on modified distance calculation (IGD+) [47], hypervolume (HV) [48], and feasible rate (FR) [49] were adopted as indicators. IGD+ measures the average distance between the true PF and the nearest individual, reflecting the algorithm's convergence. A smaller IGD+ value indicates better algorithm performance. HV calculates the volume enclosed by the obtained solutions and the predefined reference point, serving as an indicator of both convergence and diversity. A higher HV value corresponds to better algorithm performance. The FR represents the ratio of runs in which the method successfully discovers feasible solutions in the final generation. A higher FR value indicates a stronger ability to find feasible regions. Moreover, the Wilcoxon test at the 0.05 level is performed to estimate the difference between the two algorithms. "+", "-" and "≈" indicate that the compared algorithm is significantly better than, significantly worse than, or statistically similar to TSRSM.

Comparison with peer algorithms

In this part, the proposed TSRSM is compared with the 9 most state-of-the-art CMOEAs in three benchmark suits. The results of IGD+ and HV are reported in Tables 4 and 5, where the **Bold** block means the best result among the ten algorithms. To clearly analyze the results, the Wilcoxon test is conducted and the corresponding results are presented in tables. The experimental results using the IGD+ metric demonstrate that TSRSM achieves superior performance compared to CCMO, cDPEA, CMOEA_MS, DSPCMDE, TSTI, MTCMO, CMOEMT, TriP, and C3M on 21, 22, 31, 36, 30, 26, 24, 24, and 35 test functions, respectively. Similarly, according to the HV metric, TSRSM outperforms the

Table 2 Description of three benchmark test suites

Test suites	Parameter settings	Features
MW	$M = 3$, $D = 15$ is set for MW4, MW8, and MW14; $M = 2$, $D = 15$ is set for other functions.	Nonlinear constraints and diverse geometries of the CPF
LIRCMOP	$M = 3$, $D = 10$ is set for LIRCMOP13 and LIRCMOP14; $M = 2$, $D = 10$ is set for other functions	Small feasible regions
Constraint DTLZ	$M = 3$ is set for all functions; $D = 7$ is set for C1-DTLZ1, DC1-DTLZ1,and DC2-DTLZ1; $D = 12$ is set for the other functions	Multimodality, irregularity and bias

other 9 compared algorithms on 22, 22, 32, 35, 29, 26, 23, 25, and 35 test functions, respectively. This confirms the effectiveness of TSRSM.

Among those compared algorithms, CMOEMT has 6 test functions better than TSRSM on IGD+. CMOEMT uses three populations solving for the CMOP, the unconstrained MOP, and the relaxed CMOP, respectively, which is similar to TriP. Moreover, CMOEMT and TriP show similar performance on LIRCMOP5, LIRCMOP6, LIRCMOP10, and LIRCMOP11, surpassing TSRAM. The same character of those functions is that interspersed distribution of feasible and infeasible regions. The consistently searching for the UPF may be helpful in identifying a wide range of the CPF for this type of problem. Moreover, in the earlier phases, the three populations of the CMOEMT evolve independently, with one of them utilizing PPS-MOEA/D (ϵ-constrained technique), which is effective for the LIRCMOP. The same issue also arises with TriP, as it independently utilizes PPS-MOEA/D. However, when it comes to the MW and CDTLZ benchmark test sets, PPS can easily converge to a local optimum in the former phases. If PPS-MOEA/D is not effective for the current problem, it would result in wasted computational time. At the pull stage, TSRSM employs PPS-MOEA/D to mitigate the risk of local optimum, particularly considering that Pop$_2$ has already attained the UPF during the push stage. Furthermore, if the expected performance is unavailable in the pull phase, RSM will switch to the next stage. Therefore, TSRSM performs significantly better in the MW and CDTLZ problems, and slightly better in the LIRCMOP problem.

CCMO, cDPEA, and MTCMO all use two cooperative populations. The distinctive feature of MTCMO is that it enables knowledge transfer between two populations. Therefore, MTCMO is more competitive than CCMO and cDPEA. Furthermore, MTCMO shows better performance than TSRSM on MW5, MW11, MW12, and DC1_DTLZ3, which have the same feature: disconnected CPF. This may be because the knowledge transfer and improved ϵ-constraint method help MTCMO find more promising solutions at the edge of CPF for this type of problem.

DSPCMDE, TSTI, C3M, and CMOEA_MS are all multi-stage-based methods. However, they only have less than 3 functions better than TSRSM. This demonstrates the superiority of the three-stage framework in TSRSM.

Table 6 displays the FR results, demonstrating that CCMO, CMOEMT, TriP, and TSRSM achieved 100% feasible rates across three benchmark test sets. These algorithms share a common characteristic, which is the utilization of a weak coevolutionary framework. The weak coevolutionary framework employs two populations: one for the unconstrained MOP and the other for the CMOP, facilitating the population's traversal of infeasible regions. CMOEA_MS, cDPEA, DSPCMDE, TSTI, MTCMO, and C3M exhibited lower performance than TSRSM on 3, 6, 9, 1, 1, and 7 test functions, respectively.

For a visual comparison, Fig. 3 shows solutions with the median IGD+ value among 30 runs obtained by TSRSM and the compared 9 algorithms on MW10 and C1_DTLZ3. For MW10 with small and discontinuous feasible regions, TSRSM outperforms the other compared algorithms in terms of both diversity and convergence performance. DSPCMDE and C3M encounter difficulties in escaping local optimum. Other algorithms are capable of finding the CPF, but they tend to exhibit limited diversity on the CPF. TriP achieves similar convergence levels on the CPF. However, due to TriP independently employing the ϵ-constrained technique, which is susceptible to getting stuck in the local optimum for this problem type, it demonstrates inferior diversity when compared to TSRSM. For C1_DTLZ3 with multimodality, DSPCMDE, TSTI, MTCMO, and C3M fail to find feasible solutions with optimal objectives. CCMO, cDPEA, CMOEA_MS, CMOEMT, and TriP can find feasible solutions. However, they may also generate solutions that lie significantly outside the feasible region. In the case of C1_DTLZ3, the CPF coincides with the UPF, making it advantageous to search for the UPF when addressing this particular problem. TSRSM demonstrates the ability to transition to the repush stage at an early stage, leveraging knowledge transfer to effectively drive evolution. Therefore, TSRSM obtains the best diversity and convergence performance than the other algorithms.

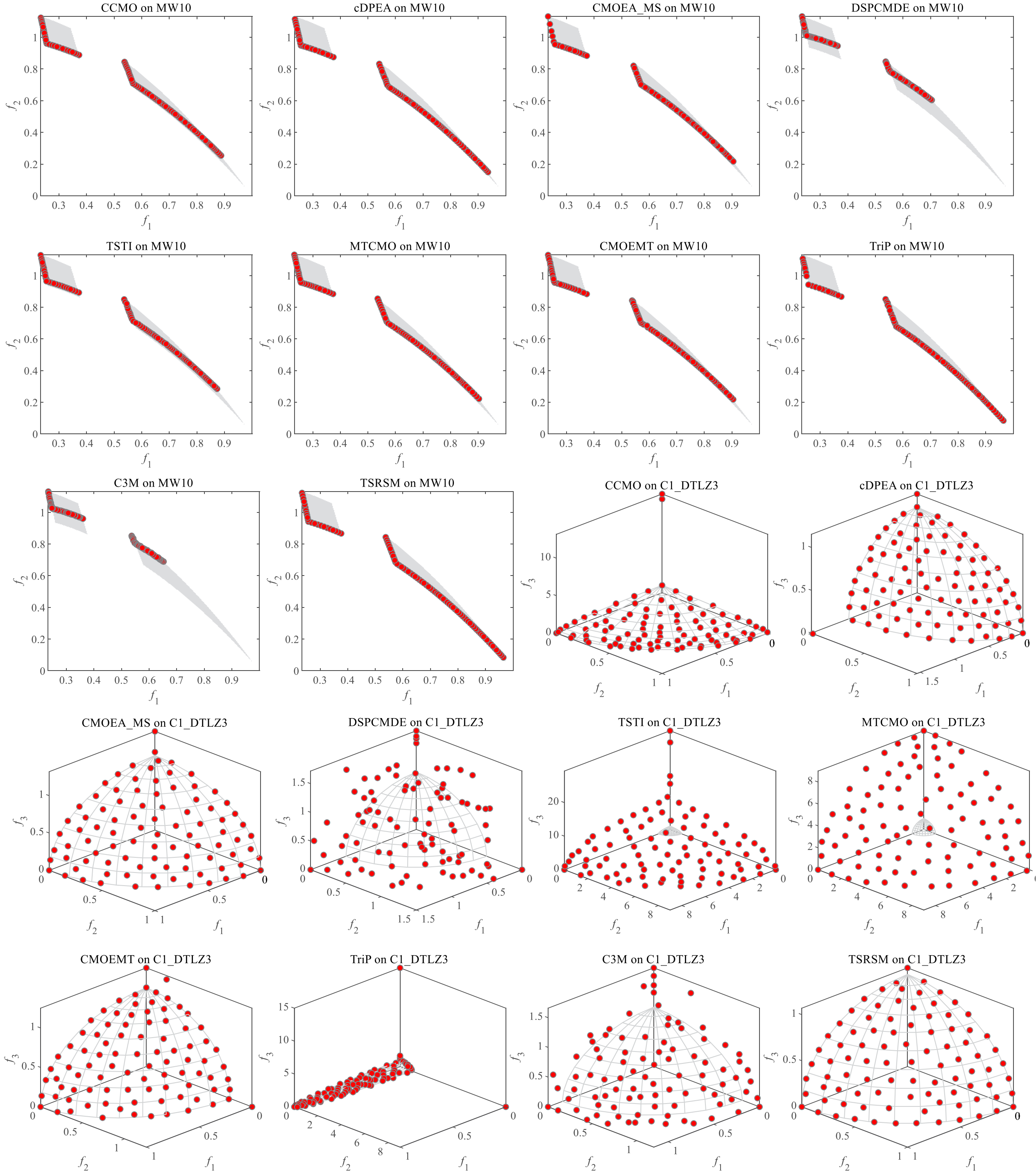


Fig. 3 Solutions with the median IGD+ value among 30 runs obtained by TSRSM and the compared 9 algorithms on MW10 and C1_DTLZ3. The red points represent solutions in Pop$_1$. The gray region is the feasible region

To further illustrate the performance of TSRSM, the average ranking of TSRSM with 9 state-of-the-art CMOEAs obtained by applying the Friedman test at the significance level 0.05, is shown in Fig. 4. The lower the ranking, the better the performance. TSRSM obtains the lowest ranking with 2.50 in HV and 2.58 in IGD+. Moreover, the rigorous difference is testified by applying post hoc statistical procedures [50]. The adjusted p-values of Friedman test with the Holm and Hochberg procedure are presented in Table 3. Therefore, TSRSM still outperforms other state-of-the-art algorithms by the Friedman test.

The exceptional performance of TSRSM can be attributed to the following factors:

1. For CDTLZ and MW which have simply feasible regions, the push and repush stage of TSRSM plays a very important role in solving this kind of problem.
2. For LIRCMOP with complex feasible regions and large infeasible regions, ϵ-constrained technique has good performance. TriP, CMOEMT, and proposed TSRSM all use ϵ-constrained technique. However, TSRSM is still better than TriP and CMOEMT, due to the cooperation between the two populations, and RSM will reward more generations for the pull stage.
3. In summary, TSRSM obtains the most competitive results in three benchmark sets with complex and uncomplicated regions. Three stages and two populations in TSRSM carry out their duties and cooperate with each other. Moreover, the RSM plays a crucial role in determining when to switch stages.

Comparison on real-world CMOPs

To provide a more comprehensive evaluation of TSRSM, real-world CMOPs are used to conduct experiments in this part. The former 30 instances of RWMOPs[21] are selected. The results of HV obtained by TSRSM and the other 9 peer algorithms are listed in Table 7. In the table, NaN indicates that the algorithm failed to find a feasible solution even after 30 runs. Despite the unique characteristics and challenges posed by different real-world CMOPs, TSRSM consistently outperforms CCMO, cDPEA, CMOEA_MS, DSPCMDE, TSTI, MTCMO, CMOEMT, TriP, and C3M on 12, 13, 19, 20, 14, 11, 15, 17, and 15 real-world CMOPs, respectively. Although CMOEMT and Trip have 8 and 5 benchmark test functions better than TSRSM on HV, respectively, they only have one or two real-world problems better than TSRSM. Furthermore, TSRSM achieves the 10 best results on real-world problems, such as the synchronous optimal pulse-width modulation of 3-level inverters problem, the multi-product batch plant problem, the heat exchanger network design problem, and others. This is the highest number of best results among the compared CMOPs. Therefore, the effectiveness of the proposed TSRSM in solving real-world problems is demonstrated.

Discussions about TSRSM

Investigation into the search behavior

We investigate the search behavior of TSRSM across different test problems. Two representative test functions, LIRCMOP1 and MW9, have been selected because they exhibit distinct sizes of feasible regions and varying degrees of overlap between the CPF and UPF. The results of IGD+, HV, RFS, and the distributions of solutions for MW9 and LIRCMOP1 are presented in Fig. 5, obtained from a single run of TSRSM. This particular run is chosen out of thirty runs due to its median IGD+.

1. The effect of using reward-switching mechanism: Fig. 5a, e depict the results of IGD+ and HV for the number of function evaluations, along with RFS in Pop_1 and Pop_2. The first switching point refers to the transition from the push stage to the pull stage, while the second switching point corresponds to the transition from the pull stage to the repush stage in Fig. 5a. Figure 5e illustrates a single switching point, specifically representing the transition from the push stage to the pull stage. From Fig. 5a, e, we can obtain the following observations:

1. The RSM will transition to the next stage when the process of evolution becomes significantly slow or stagnant.
2. If a particular stage consistently performs well on the problem, there would be no need to switch to another stage, such as LIRCMOP1 depicted in Fig. 5e.

To further showcase the competitiveness of the RSM, two additional popular switching mechanisms are employed in TSRSM. For the number of evaluations method, when $\alpha = 0.2$, for most problems, the Pop_2 could reach the UPF [33]. Therefore, to simplify the experiment, α and β are set to 0.2 and 0.4, respectively, in TSRSM1. Another switching mechanism used in TSRSM2 is the gradient of the maximum of the nadir and the minimum of ideal points [5, 28, 37], which is employed to evaluate the performance of the population. The RFS method is not applicable to TSRSM since it is specifically designed for dual-stage methods. Furthermore, the switching mechanism utilized in C3M is not applicable due to its implementation of gradually increasing constraint numbers during the middle stage. In C3M, when the constraint numbers meet the threshold, the final stage is activated. Therefore, only two switching mechanisms are selected to compare with RSM. However, those two switching mechanisms are popularly used in many algorithms as discussed in "Related works and motivation".

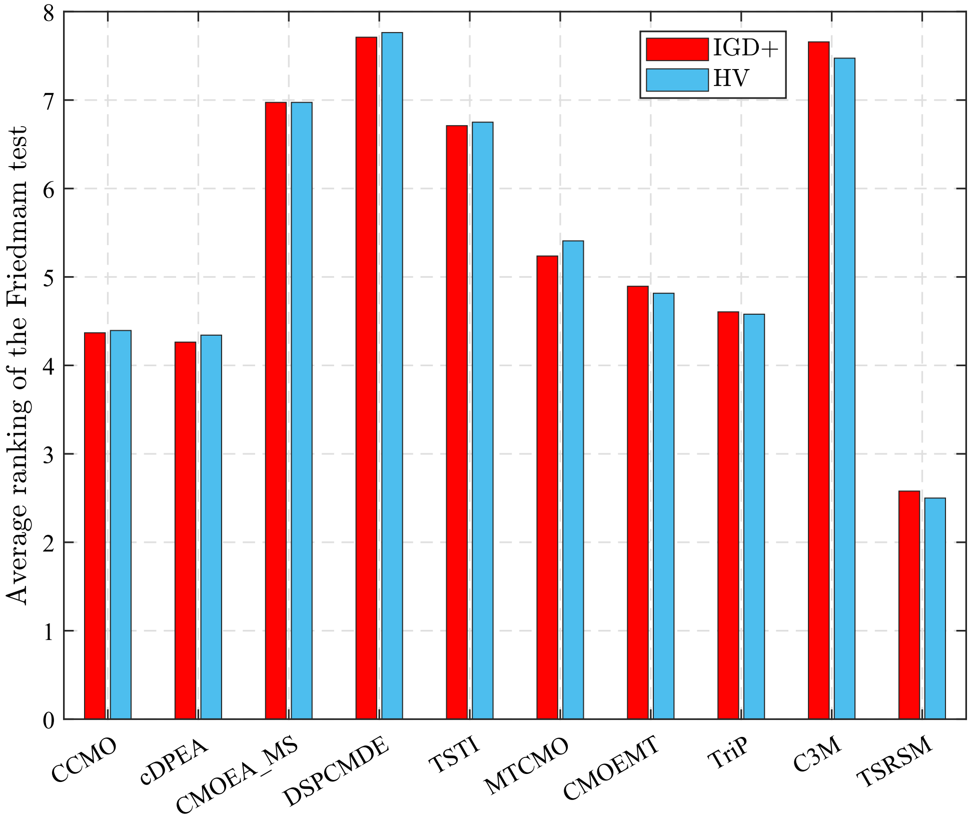


Fig. 4 Averge ranking of all 10 algorithms on three benchmark test sets by Friedman test

Table 3 The adjusted p-values of IGD+ and HV obtained through Friedman test of TSRSM and 9 state-of-the-art algorithms

TSRSM VS	IGD+		HV	
	Holm p	Hochberg p	Holm p	Hochberg p
CCMO	0.019973	0.015319	0.012750	0.008000
cDPEA	0.019973	0.015319	0.012750	0.008000
CMOEA_MS	0.000000	0.000000	0.000000	0.000000
DSPCMDE	0.000000	0.000000	0.000000	0.000000
TSTI	0.000000	0.000000	0.000000	0.000000
MTCMO	0.000650	0.000650	0.000142	0.000142
CMOEMT	0.003424	0.003424	0.003424	0.003424
TriP	0.010593	0.010593	0.008286	0.008000
C3M	0.000000	0.000000	0.000000	0.000000

The detailed IGD+ results and HV results of those three algorithms are listed in Table 8 and Table 9. Comparison results demonstrate that TSRSM outperforms TSRSM1 and TSRSM2 on 11 and 9 test functions, respectively, according to IGD+ results. Similarly, TSRSM achieves competitive better results on HV.

2. The effect of using two populations: To investigate the roles of the two populations, the distributions of solutions are monitored at the switching point and the final evaluation in Fig. 5b–d and f–h, in addition to the RFS in the different populations (Fig. 5a, e. From Fig. 5, we have the following observations:

1. Pop_1 mainly serves as storing feasible solutions, while Pop_2 has the ability to preserve both infeasible and feasible solutions simultaneously. That is, Pop_1 aims to search for the CPF, thus focusing on the feasible regions. Pop_2 aims to converge to the UPF on the push stage and the repush stage, and toward to the CPF on the pull stage. Furthermore, the RFS in Pop_2 varies depending on the geometry of the feasible and infeasible regions. As is shown in Fig. 5e, the RFS exhibits significant fluctuations in the pull stage for LIRCMOP1 due to its small feasible regions. In contrast, MW9 showcases minimal fluctuations in the pull stage given its larger feasible regions in Fig. 5a.

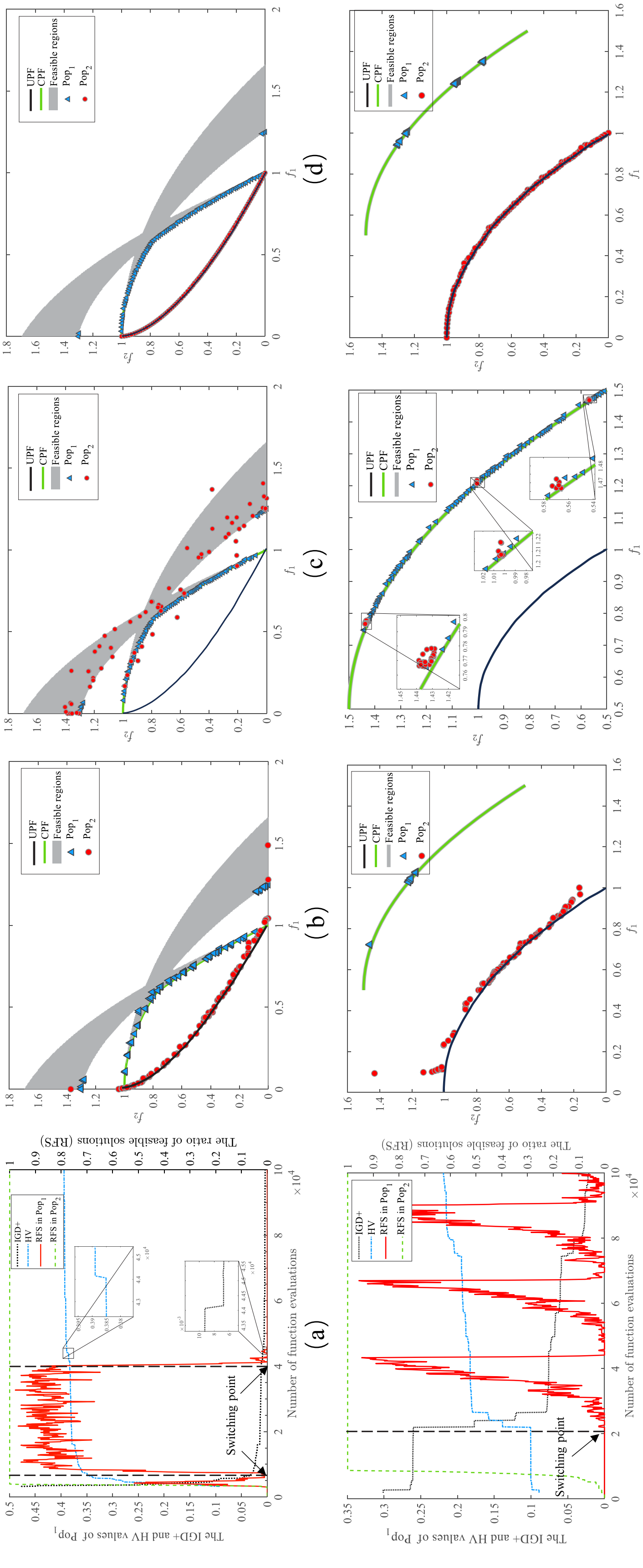


Fig. 5 The IGD+, HV, and RFS results of TSRSM on MW9 and LIRCMOP1 are, respectively, shown in **a** and **e**. **b–d** present the distributions of solutions on MW9 at the first and second switching points, as well as the final evaluation, respectively. **f, g** Show the distributions of solutions on LIRCMOP1 at the first switching point and the second switching point. **h** Shows the distributions of solutions on LIRCMOP1 at the final evaluation for TSRSM without the pull stage (TSRSM-PR)

2. In the push stage, Pop_2 converges to the UPF, which has better objective values than Pop_1, as presented in Fig. 5b and Fig. 5f. Therefore, Pop_2 can help Pop_1 cross infeasible regions.
3. In the pull stage, Pop_2 utilizes the ϵ-constraint method [28], resulting in an initial increase in RFS. Once the RFS exceeds a threshold value φ (e.g., 0.95), the constraint is then relaxed, leading to oscillations in the RFS of Pop_2. In other words, when RFS exceeds φ, the weights of constraints are decreased to preserve population diversity. Conversely, if the RFS is below the threshold, the weights of constraints are increased to guide the population toward convergence to the CPF.
4) In the repush stage, Pop_2 undergoes knowledge transfer with Pop_1, when moving toward the UPF once more, thereby aiding in the expansion of the unexplored CPF for Pop_1.

3. The effect of using tri-stage: The proposed TSRSM has three stages: the push stage, the pull stage, and the repush stage. Therefore, Three variants are designed for the algorithm. First, we discuss the different order between the pull stage and the repush stage. When the repush stage is before the pull stage, the algorithm is named TSRSM-PRP. Then, another two variants of TSRSM are considered: TSRSM-PR, which excludes the pull stage, and TSRSM-PP, which excludes the repush stage.

The detailed IGD+ and HV results of those three algorithms are reported in Table 10 and Table 11. TSRSM outperforms TSRSM-PR, TSRSM-PP, and TSRSM-PRP in terms of IGD+ and HV results. TSRSM-PRP exhibits poorer performance compared to TSRSM-PR, TSRSM-PP, and TSRSM, indicating that the pull stage should precede the repush stage. TSRSM-PP has better performance on LIRCMOP than other methods. This implies that the ϵ-constrained technique in the pull stage is more effective in passing through infeasible regions and solving problems with small feasible regions. As presented in Fig. 5h, TSRSM-PR obtains poor diversity on the CPF, while TSRSM-PP gets a well-distributed set of solutions in Fig. 5g. However, the ϵ-constrained technique is susceptible to falling into local optimum easily. Figure 6 illustrates the performance of TSRSM-PP and TSRSM on MW13. MW13 exhibits the characteristics of slender and narrow feasible regions, and is prone to local optimum. TSRSM-PP displays a less favorable distribution on the CPF when compared to TSRSM. This discrepancy arises from the fact that TSRSM has the capability to switch to the repush stage, which increases the likelihood of escaping the local optimum. Therefore, the tri-stage of TSRSM is the most effective.

Investigation into the main strategies

In this section, four variants are designed to verify the effectiveness of proposed strategies in TSRSM. The first variant (TSRSM-V1) is to verify the effectiveness of using Pop_1 to calculate sk and sd. Therefore, the first variant employs Pop_2 to calculate sc_k and sd_k. The second variant (TSRSM-V2) employs 20 generations to reward the two preceding stages, while the third variant (TSRSM-V3) utilizes 200 generations, thus confirming the effectiveness of varying the number of rewarded generations in RSM. The fourth variant (TSRSM-V4) uses offspring Off_2 to attend the CP_1, not Pop_2, which is used to verify the effectiveness of the cooperation of two populations on the pull stage.

Table 12 shows the performance of TSRSM and four variants on LIRCMOP problems. TSRSM is significantly better than TSRSM-V1, TSRSM-V2, TSRSM-V3, and TSRSM-V4 on 4, 9, 8, and 11 problems, respectively. Moreover, TSRSM obtains 7 best average values on LIRCMOP problems. Hence, the effectiveness of the proposed strategies in TSRSM can be verified.

Parameter analysis of TSRSM

The proposed TSRSM contains three parameters λ, gr_1 and gr_2 to adjust the RSM. The setting of gr_1 and gr_2 is dependent on λ. In other words, the values of gr_1 and gr_2 vary when λ takes different values. Due to space limitations, we only present the effects of gr_1 and gr_2 when $\lambda = 1\text{e–}2$, despite conducting numerous experiments. Here are the experimental details.

1. TSRSM3: $gr_1 = 0$ and $gr_2 = 200$.
2. TSRSM4: $gr_1 = 40$ and $gr_2 = 200$.
3. TSRSM5: $gr_1 = 60$ and $gr_2 = 200$.
4. TSRSM6: $gr_1 = 80$ and $gr_2 = 200$.
5. TSRSM7: $gr_1 = 20$ and $gr_2 = 0$.
6. TSRSM8: $gr_1 = 20$ and $gr_2 = 400$.
7. TSRSM9: $gr_1 = 20$ and $gr_2 = 600$.
8. TSRSM10: $gr_1 = 20$ and $gr_2 = 800$.

The comparison results in terms of IGD+ and HV are presented in Tables 13, 14, 15 and 16. When $gr_1 = 0$ or $gr_2 = 0$, TSRSM obtains very poor results, indicating that both the push stage and the pull stage are crucial for TSRSM. TSRSM4 (with $gr_1 = 40$ and $gr_2 = 200$) and TSRSM5 (with $gr_1 = 60$ and $gr_2 = 200$) exhibit comparable performance to TSRSM (with $gr_1 = 20$ and $gr_2 = 200$), suggesting that varying gr_1 from 20 to 60 has negligible impact on the performance of TSRSM. As gr_1 and gr_2 increase, the number of generations left in TSRSM becomes smaller, which means that TSRSM has less chance to make adjustments. In other words, the larger gr_1 and gr_2, the worse performance. Moreover, among the 9 variants of TSRSM with different values for gr_1 and gr_2, TSRSM with $gr_1 = 20$ and $gr_2 = 200$ demonstrates superior performance. Therefore, $gr_1 = 20$ and $gr_2 = 200$ are set in this paper.

Conclusions and future work

This paper proposes a tri-stage with reward-switching mechanism framework named TSRSM for CMOPs. The tri-stage consists of the push, pull, and repush stage. Each stage consists of two coevolutionary populations, namely Pop_1 and Pop_2. Among the two populations, Pop_1 is dedicated to converging toward the CPF over three stages, whereas Pop_2 is designed to go back and forth between the UPF and CPF. Moreover, RSM is applied to determine when to switch stages according to the maximum rate of change between the convergence and diversity in the population. The experimental results on three benchmark test sets and 30 real-world CMOPs demonstrate that TSRSM outperforms 9 state-of-the-art peer CMOEAs. Furthermore, TSRSM demonstrates competitive performance in addressing problems with complex infeasible regions and small discrete feasible regions, such as the synchronous optimal pulse-width modulation of 3-level inverters problem.

However, the current TSRSM still has certain limitations, which can be addressed through the following potential approaches.

1. TSRSM still performs poorly on some test functions such as LIRCMOP5-6, and LIRCMOP10-11. TSRSM-PP and TSRSM4 with a large proportion of the pull stage perform better in these test functions, which means that it is necessary to design a better RSM for improving the accuracy of the switching. The integration of RSM with machine learning techniques, which can learn the characteristics of various stages and problems, shows potential in determining optimal switching strategies.
2. TSRSM requires setting three parameters that significantly impact the algorithm's performance. To simplify the application and enhance efficiency, advanced techniques can be employed to minimize the number of required parameters. Furthermore, utilizing the concept of iterative learning control may offer a feasible approach to dynamically tune the parameters of TSRSM [51, 52].

The following research directions are suggested for future study.

1. Further research is warranted for fuzzy-based TSRSM to address the challenges presented by real-world multiobjective optimization problems with uncertain parameters and uncertain semantic representations [53–55].
2. It is worth considering the exploration of adaptive selection of different evolutionary operators in TSRSM.

Acknowledgements This work was supported by the Shanghai Science and Technology Planning Project (20DZ2205900), Shanghai Municipal Commission of Science and Technology Project (19511132101), Shanghai Municipal Science and Technology Major Project (2021SHZD ZX0100), and the Fundamental Research Funds for the Central Universities.

Data availability The data that support the findings of this study are openly available.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

Appendix A Table

See Tables 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16.

Table 4 The mean IGD+ values of ten algorithms on three sets of benchmark test functions

Problem	CCMO	cDPEA	CMOEA_MS	DSPCMDE	TSTI	MTCMO	CMOEMT	TriP	C3M	TSRSM
C1_DTLZ1	1.4861e–2 –	1.4831e–2 –	1.5303e–2 –	2.2574e–2 –	1.4832e–2 –	1.4833e–2 –	1.6680e–2 –	1.6009e–2 –	1.9881e–2 –	**1.4700e–2**
C1_DTLZ3	2.3930e–2 ≈	**2.3668e–2** ≈	2.3960e–2 ≈	6.4315e–1 –	4.2916e+0 –	5.8850e+0 –	2.9594e–2 –	2.6589e–2 –	5.1427e–1 –	2.3835e–2
C2_DTLZ2	2.0147e–2 –	2.0637e–2 –	**1.9261e–2** ≈	3.0226e–2 –	1.9745e–2 –	1.9938e–2 –	2.1403e–2 –	2.1121e–2 –	3.1382e–2 –	1.9268–2
C3_DTLZ4	6.9026e–2 ≈	6.1636e–2 –	3.4524e–1 –	9.2492–2 –	5.8852e–2 ≈	5.8731e–2 ≈	6.7198e–2 –	6.6343–2 –	1.0074e–1 –	**5.8194e–2**
DC1_DTLZ1	8.5298e–3 ≈	8.5409e–3 –	1.1693e–2 –	2.8341e–1 –	8.7462e–3 –	8.5714e–3 –	1.0419e–2 –	9.8758e–3 –	2.8453e–2 –	**8.4602e–3**
DC1_DTLZ3	1.3606e–2 ≈	1.3572e–2 +	3.8554–2 ≈	6.0785e–1 –	2.0150e–2 ≈	**1.3522e–2** +	1.4556e–2 ≈	1.8479-2 –	6.9195e–1 –	1.7325e–2
DC2_DTLZ1	1.5014e–2 –	1.4995e–2 –	1.5419-2 –	2.5900e–2 –	4.0788e–2 –	9.7683e–2 –	1.6085e–2 –	1.7360e–2 –	2.7777e–2 –	**1.4953e–2**
DC2_DTLZ3	**2.3672e–2** ≈	1.4995e–1 –	4.5351e–1 –	5.6584e–1 –	5.6113e–1 ≈	5.6209e–1 –	2.9277e–2 ≈	1.1476e–1 –	2.8800-1 –	1.0198e–1
DC3_DTLZ1	**5.1612e–3** ≈	5.1796-3 ≈	2.4734-2 –	3.7114e–1 –	2.2038e–1 –	9.2825e–2 –	5.4102-3 ≈	6.0637e–3 +	2.7949e+0 –	1.0792e–2
DC3_DTLZ3	**8.2650-3** +	3.0117e–1 ≈	1.3629e–1 +	6.3396-1 –	1.8555e+0 –	1.1624e+0 –	9.1676e–3 +	2.5910e–1 –	2.0984e+0 –	1.7186e–1
LIRCMOP1	1.7740-1 –	9.5262e–2 –	2.6350-1 –	4.7940e–2 –	9.5253e–2 –	1.4413e–2 ≈	**1.3511e–2** ≈	1.4469e–2 ≈	8.5188-2 –	1.4963e–2
LIRCMOP2	8.6248e–2 –	6.1034-2 –	2.0419-1 –	4.6575e–2 –	7.4617-2 –	1.5327-2 –	**8.7778e–3** ≈	9.7149e–3 –	7.4719-2 –	9.0102e–3
LIRCMOP3	2.3367-1 –	1.520-1 –	2.6465e–1 –	9.3501e–2 –	1.9182-1 –	1.2846-1 –	1.4453e–2 ≈	6.2949e–2 ≈	3.0549e–1 –	**1.4317e–2**
LIRCMOP4	1.3725-1 –	6.8437-2 –	2.1138e–1 –	8.4479-2 –	6.9215-2 –	1.2128e–2 ≈	1.3294-2 ≈	**9.7424e–3** ≈	1.1548e–1 –	1.5124-2
LIRCMOP5	1.2021e–2 –	1.7466-2 –	2.7121e–2 –	6.0131-3 –	1.0781e–1 –	6.4338-2 –	**3.2907e–3** +	3.3102e–3 +	4.9344e–3 +	5.6297-3
LIRCMOP6	9.0318e–3 –	1.8887-2 –	3.1702-2 –	5.6632e–3 –	1.4829e–1 –	1.0835e–1 –	**2.9113e–3** +	2.9642e–3 +	4.7341e–3 –	3.3155-3
LIRCMOP7	1.0276e–2 –	8.9782-3 –	1.3768e–2 –	7.5692e–3 –	2.2035-2 –	1.3234e–2 –	7.1476-3 –	6.9167e–3 –	6.8196-3 –	**6.5277e–3**
LIRCMOP8	7.2106e–3 –	7.2070-3 –	1.3945e–2 ≈	7.5213e–3 –	5.1494-2 –	8.7616e–3 ≈	7.1400-3 –	6.7855e–3 –	6.6520-3 –	**6.4708e–3**
LIRCMOP9	1.6485e–2 ≈	1.6217e–2 ≈	2.8361-1 –	3.5849e–2 –	2.6441-1 –	6.9404e–2 –	1.5985-2 ≈	2.5554-2 ≈	**2.5870e–3** +	2.2200-2
LIRCMOP10	5.3306e–3 –	9.2608-3 –	5.4768e–2 –	8.1776-3 –	2.9715e–1 –	2.7814e–2 –	**3.3717-3** +	3.4405e–3 +	4.3049e–3 –	4.1256-3
LIRCMOP11	6.5560e–4 ≈	**6.2737e–4** ≈	6.5833e–2 –	7.6675e–4 +	1.7959-1 –	4.5533-3 –	8.1643e–4 +	8.6951e–4 +	6.2783-4 ≈	2.3187e–3
LIRCMOP12	1.3257-3 ≈	**4.8669e–4** ≈	5.4565-2 –	1.1714e–3 –	1.5793e–1 –	1.5536-2 –	1.6405e–3 –	8.1430e–4 –	8.3930e–4 –	7.9006-4
LIRCMOP13	4.6656e–2 –	4.5892-2 –	4.5313-2 –	7.1111e–2 –	4.6375e–2 –	4.7087-2 –	4.6949e–2 –	4.8052-2 –	7.1873e–2 –	**4.2186e–2**
LIRCMOP14	4.9194-2 –	4.8540e–2 ≈	**4.7348e–2** +	6.1675-2 –	4.7944e–2 ≈	4.8378-2 ≈	5.0613e–2 –	5.0223-2 –	6.2441e–2 –	4.8073e–2
MW1	1.2773-3 –	1.2687e–3 –	2.5875-3 –	1.2327e–2 –	1.9626-3 –	1.6163e–3 –	1.4311-3 –	1.4885e–3 –	3.9828-2 –	**1.2487e–3**
MW2	2.1842e–2 ≈	2.0490-2 ≈	2.6083e–2 –	5.3695-2 –	2.3958e–2 ≈	2.3771e–2 –	2.7079-2 –	1.8696-2 ≈	9.0166e–2 –	**1.7687e–2**
MW3	3.1666e–3 –	9.9665-3 –	3.4271e–3 –	1.0112-2 –	2.9821e–3 ≈	3.2040-3 –	3.3782e–3 –	3.3730-3 –	4.2612-3 –	**2.9021e–3**
MW4	3.0648e–2 –	3.6903e–2 –	3.1324-2 –	7.6036e–2 –	3.0458e–2 –	3.0637-2 –	3.5396e–2 –	4.1698e–2 –	4.8036-2 –	**2.9735e–2**
MW5	1.0803e–3 ≈	1.4889e–1 –	2.5977-2 –	1.6475e–1 –	4.7660e–2 –	**5.5336e–4** +	2.3325-3 +	1.2875e–1 –	8.2682-2 –	6.6809e–3
MW6	2.9123e–2 –	3.2695-2 ≈	6.2033e–2 –	2.6819-1 –	6.3290e–2 –	3.8855-2 ≈	7.1580e–2 –	1.7797-2 ≈	4.0028e–1 –	**1.6864e–2**
MW7	**2.0862-3** ≈	2.1752-3 ≈	4.2875e–3 –	2.1288-3 ≈	2.5665e–3 ≈	2.2868-3 –	2.6783e–3 –	2.5747-3 –	2.8030e–3 –	2.0949-3
MW8	2.8995e–2 ≈	2.7256e–2 ≈	3.6138-2 –	8.9856e–2 –	3.6708-2 –	2.9622e–2 ≈	3.1461-2 –	**2.6293e–2** ≈	1.0733-1 –	2.7142e–2
MW9	3.3086-3 ≈	**3.2901e–3** ≈	1.0579e–1 –	2.4741e–1 –	5.8428-2 –	3.3049-3 ≈	3.9783e–3 –	5.4070-3 –	1.6982e–1 –	3.3726-3
MW10	3.5181e–2 –	**1.7899e–2** ≈	5.6203e–2 –	1.6153-1 –	9.8854e–2 –	4.2814-2 –	6.6398e–2 –	1.8411-2 ≈	2.7549e–1 –	2.2101-2
MW11	2.8499e–3 +	3.0869e–3 –	3.8383e–2 –	4.3563e–1 –	2.8738-3 +	**2.7753e–3** +	3.5198e–3 –	3.3547e–3 –	4.1315-3 –	2.9800e–3
MW12	3.3483e–3 ≈	3.4321e–3 ≈	3.2442e–3 +	3.4931-3 –	3.9331e–3 –	**3.0949-3** +	2.5078e–2 –	4.5982-3 –	2.6619e–1 –	3.3639-3
MW13	4.9322e–2 –	2.0252-2 ≈	7.0291-2 –	1.2881e–1 –	8.1942e–2 –	4.3416e–2 –	6.6101e–2 –	**2.0145e–2** ≈	1.9138e–1 –	2.3501-2
MW14	6.9695e–2 –	6.7716-2 –	8.2397e–2 –	8.7935e–2 –	6.6725e–2 –	6.7091-2 –	7.1684-2 –	7.4256e–2 –	8.7176e–2 –	**6.4823-2**
+/ – / ≈	2/21/15	1/22/15	3/31/4	1/36/1	1/30/7	4/26/8	6/24/8	5/24/9	2/35/1	

Table 5 The mean HV values of ten algorithms on three sets of benchmark test functions

Problem	CCMO	cDPEA	CMOEA_MS	DSPCMDE	TSTI	MTCMO	CMOEMT	TriP	C3M	TSRSM
C1_DTLZ1	8.3964e–1 –	8.3961-1 –	8.3646e–1 –	8.0932-1 –	8.4066e–1 –	8.3912-1 –	8.3487-1 –	8.3390e–1 –	8.1851-1 –	**8.4144e–1**
C1_DTLZ3	5.5809-1 –	5.5836e–1 ≈	5.5827e–1 ≈	6.4877-2 –	2.4907-1 –	1.4856e–1 –	5.4829-1 –	5.5331e–1 –	3.4258-1 –	**5.5858e–1**
C2_DTLZ2	5.1426-1 –	5.1360-1 –	5.1533e–1 –	4.8070e–1 –	5.1487-1 –	5.1418e–1 –	5.0971e–1 –	5.1379-1 –	4.8936-1 –	**5.1711e–1**
C3_DTLZ4	7.8002e–1 ≈	7.8601-1 –	4.9996e–1 –	7.6365e–1 –	7.8773-1 ≈	7.8776e–1 ≈	7.8240e–1 –	7.8305e–1 –	7.5675e–1 –	**7.8815-1**
DC1_DTLZ1	6.3197e–1 ≈	6.3178e–1 –	6.1284e–1 –	2.9783-1 –	6.3120e–1 –	6.3151e–1 –	6.2514-1 –	6.2816e–1 –	5.6556e–1 –	**6.3224e–1**
DC1_DTLZ3	4.7309-1 ≈	**4.7336e–1** +	4.5718e–1 ≈	2.7816e–2 –	4.6831-1 ≈	4.7311e–1 ≈	4.7037e–1 +	4.6352e–1 –	1.6342e–1 –	4.6630-1
DC2_DTLZ1	8.4045e–1 –	8.4041e–1 –	8.3797-1 –	8.1130e–1 –	7.7747-1 –	6.3865e–1 –	8.3782e–1 –	8.3501e–1 –	8.1062e–1 –	**8.4093e–1**
DC2_DTLZ3	**5.5847e–1** ≈	4.3045e–1 –	1.2238e–1 –	2.6751-2 –	1.2940-2 ≈	1.3396e–2 –	5.4732-1 +	4.2862-1 –	2.7990e–1 –	4.8035-1
DC3_DTLZ1	**5.3568e–1** ≈	5.3540e–1 ≈	4.0999-1 –	1.9432e–1 –	1.2655e–1 –	3.0343-1 –	5.3484e–1 ≈	5.3208-1 +	9.7425e–2 –	5.2057e–1
DC3_DTLZ3	**3.6775e–1** +	1.7149e–1 ≈	2.8180-1 +	1.8316e–2 –	0.0000e+0 –	0.0000e+0 –	3.6527-1 +	1.7633e–1 –	4.1074-2 –	2.5323e–1
LIRCMOP1	1.4796e–1 –	1.8380-1 –	1.1707e–1 –	2.1268-1 –	1.7779e–1 –	2.2983e–1 ≈	**2.3158e–1** ≈	2.3092e–1 ≈	1.8626-1 –	2.3071e–1
LIRCMOP2	2.8695-1 –	3.1688e–1 –	2.1356-1 –	3.2838e–1 –	3.0083e–1 –	3.5170-1 –	**3.5751e–1** ≈	3.5683-1 –	2.9980e–1 –	3.5740-1
LIRCMOP3	1.1083e–1 –	1.3806-1 –	9.6611-2 –	1.6646e–1 –	1.2323-1 –	1.4981e–1 –	1.9295e–1 ≈	1.7649-1 ≈	8.2040-2 –	**1.9929e–1**
LIRCMOP4	2.2302-1 –	2.6946e–1 –	1.8465-1 –	2.6232e–1 –	2.6773e–1 –	3.0978-1 ≈	3.0791e–1 ≈	**3.1074-1** ≈	2.3948e–1 –	3.0688-1
LIRCMOP5	2.8538e–1 –	2.7793-1 –	2.6939e–1 –	2.9139-1 +	2.2668-1 –	2.6912e–1 –	**2.9276-1** +	2.9266e–1 +	2.9194-1 +	2.9068e–1
LIRCMOP6	1.9487e–1 –	1.9092e–1 –	1.8575-1 –	1.9661e–1 –	1.3615e–1 –	1.7246-1 –	**1.9810-1** +	1.9807e–1 +	1.9712e–1 –	1.9790-1
LIRCMOP7	2.9101e–1 –	2.9134e–1 –	2.8926e–1 –	2.9362e–1 –	2.8597-1 –	2.8910e–1 –	2.9388e–1 –	2.9390e–1 –	2.9399-1 –	**2.9415e–1**
LIRCMOP8	2.9317e–1 –	2.9311e–1 –	2.8985-1 ≈	2.9365-1 –	2.8046e–1 –	2.9187e–1 ≈	2.9390-1 –	2.9401e–1 –	2.9405e–1 –	**2.9418e–1**
LIRCMOP9	5.5871e–1 ≈	5.4280-1 –	4.2112e–1 –	5.2958e–1 –	3.7856-1 –	5.3145-1 –	5.5930e–1 ≈	5.5528-1 –	**5.6675e–1** +	5.5690e–1
LIRCMOP10	7.0665-1 –	6.9545-1 –	6.7774e–1 –	6.9589e–1 –	5.0244-1 –	6.9050-1 –	**7.0801e–1** +	7.0789-1 +	7.0728e–1 –	7.0776-1
LIRCMOP11	6.9393e–1 ≈	**6.9397e–1** +	6.5061e–1 –	6.9383-1 +	5.4928e–1 –	6.9089-1 –	6.9385-1 +	6.9381e–1 +	6.9397-1 +	6.9282e–1
LIRCMOP12	6.1986-1 ≈	**6.2027e–1** +	5.9223-1 –	6.1964e–1 –	5.2267e–1 –	6.1234-1 –	6.1964-1 –	6.2009e–1 –	6.2001-1 –	6.2012e–1
LIRCMOP13	5.5216e–1 –	5.5285-1 –	5.5324e–1 –	5.1804-1 –	5.5257e–1 –	5.5139-1 –	5.5149e–1 –	5.5026e–1 –	5.2635e–1 –	**5.5715-1**
LIRCMOP14	5.5092e–1 –	5.5153e–1 ≈	**5.5250e–1** ≈	5.3014e–1 –	5.5215e–1 ≈	5.5162e–1 ≈	5.4897e–1 –	5.4958e–1 –	5.3887e–1 –	5.5201e–1
MW1	4.8998e–1 ≈	**4.9002e–1** +	4.8737-1 –	4.7643e–1 –	4.8782e–1 ≈	4.8889e–1 ≈	4.8961e–1 –	4.8949e–1 –	4.4106e–1 –	4.9000e–1
MW2	5.5059e–1 ≈	5.5322e–1 ≈	5.4427e–1 –	5.0573e–1 –	5.4760e–1 ≈	5.4758e–1 –	5.4321e–1 –	5.5572e–1 ≈	4.5808-1 –	**5.5704e–1**
MW3	5.4400e–1 –	5.3191e–1 –	5.4361-1 –	5.3173e–1 –	5.4434e–1 ≈	5.4393e–1 –	5.4353e–1 –	5.4355e–1 –	5.4214e–1 –	**5.4442e–1**
MW4	8.3989e–1 –	8.3187e–1 –	8.3756-1 –	7.8036e–1 –	8.3998e–1 –	8.3996e–1 –	8.3431e–1 –	8.2669e–1 –	8.1987e–1 –	**8.4110-1**
MW5	3.2412-1 ≈	1.8929e–1 –	3.0098e–1 –	1.7359e–1 –	2.8785-1 –	**3.2430e–1** +	3.2339-1 +	2.0981e–1 –	2.4819-1 –	3.1947e–1
MW6	2.9142-1 –	2.9630e–1 ≈	2.7673e–1 –	1.7997-1 –	2.8405e–1 –	2.9298-1 ≈	2.7564e–1 –	3.0143-1 ≈	1.3397e–1 –	**3.0690e–1**
MW7	4.1225e–1 ≈	4.1221e–1 ≈	4.0890e–1 –	**4.1247e–1** +	4.1154-1 ≈	4.1172e–1 –	4.1142e–1 –	4.1169e–1 –	4.1097-1 –	4.1224e–1
MW8	5.3431e–1 ≈	5.3729-1 ≈	5.2119e–1 –	4.1510e–1 –	5.2019e–1 –	5.3320e–1 –	5.2996e–1 –	**5.3958-1** ≈	3.9609e–1 –	5.3830e–1
MW9	3.9791e–1 ≈	**3.9895e–1** +	3.0151e–1 –	1.6006e–1 –	3.5999e–1 –	3.9680e–1 ≈	3.9665e–1 ≈	3.9478e–1 –	2.8407e–1 –	3.9740e–1
MW10	4.1873e–1 –	4.3418e–1 ≈	4.0423e–1 –	3.3214e–1 –	3.7519e–1 –	4.1307e–1 –	3.9760e–1 –	**4.3450e–1** ≈	2.6553e–1 –	4.3043e–1
MW11	4.4715e–1 –	4.4742-1 –	4.3547e–1 ≈	3.1024e–1 –	4.4750e–1 ≈	**4.4775e–1** +	4.4727-1 –	4.4719e–1 –	4.4664e–1 –	4.4758e–1
MW12	6.0432e–1 ≈	6.0425e–1 ≈	6.0372e–1 –	6.0361e–1 –	6.0329e–1 –	**6.0472-1** +	5.8410e–1 –	6.0249e–1 –	3.7279-1 –	6.0419-1
MW13	4.3960e–1 –	4.6168-1 ≈	4.2652e–1 –	3.8792-1 –	4.2124e–1 –	4.4420e–1 –	4.3117e–1 –	**4.6187e–1** ≈	3.5757-1 –	4.5873e–1
MW14	4.7041e–1 –	4.7113e–1 –	4.6570e–1 –	4.4831e–1 –	4.7191-1 –	4.7206e–1 –	4.6795e–1 –	4.6783e–1 –	4.5325e–1 –	**4.7407e–1**
+/ – / ≈	1/22/15	5/22/11	1/32/5	3/35/0	0/29/9	3/26/9	8/23/7	5/25/8	3/35/0	

Table 6 The FR values of ten algorithms on three sets of benchmark test functions

Problem	CCMO	cDPEA	CMOEA_MS	DSPCMDE	TSTI	MTCMO	CMOEMT	TriP	C3M	TSRSM
C1_DTLZ1	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	2.6667e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
C1_DTLZ3	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
C2_DTLZ2	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
C3_DTLZ4	1.0000e+0 ≈	1.0000e+0 ≈	1.8571e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.7143e–1 ≈	1.0000e+0
DC1_DTLZ1	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
DC1_DTLZ3	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
DC2_DTLZ1	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	2.3333e–1 –	9.6667e–1 ≈	9.6667e–1 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.6667e–1 ≈	1.0000e+0
DC2_DTLZ3	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	6.3333-1 ≈	3.3333e–2 –	2.0000e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	9.3333e–1 ≈	9.3333e–1
DC3_DTLZ1	1.0000e+0 ≈	1.0000e+0 ≈	3.4469-1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	6.0586e–1 –	1.0000e+0
DC3_DTLZ3	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.3333e–1 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	8.2967e–1 –	1.0000e+0
LIRCMOP1	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP2	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.3480e–1 ≈	1.0000e+0
LIRCMOP3	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.6484e–1 ≈	1.0000e+0
LIRCMOP4	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	8.3407e–1 –	1.0000e+0
LIRCMOP5	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP6	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP7	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP8	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP9	1.0000e+0 ≈	2.7912e–1 –	1.0000e+0 ≈	2.6337e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP10	1.0000e+0 ≈	5.1612e–1 –	1.0000e+0 ≈	5.0366e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP11	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP12	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
LIRCMOP13	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.9377e–1 ≈	1.0000e+0
LIRCMOP14	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.6227e–1 ≈	1.0000e+0

Table 6 continued

Problem	CCMO	cDPEA	CMOEA_MS	DSPCMDE	TSTI	MTCMO	CMOEMT	TriP	C3M	TSRSM
MW1	1.0000e+0 ≈	9.6667e–1 ≈	9.3333e–1 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.6667e–1 ≈	1.0000e+0 ≈	5.6667-1 –	1.0000e+0
MW2	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW3	1.0000e+0 ≈	5.4652e–1 –	1.0000e+0 ≈	5.4322-1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW4	1.0000e+0 ≈	7.6777e–1 –	1.0000e+0 ≈	4.4945e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW5	1.0000e+0 ≈	2.4176e–2 –	9.6667e–1 ≈	3.7363e–2 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	2.9853e–1 –	1.0000e+0
MW6	1.0000e+0 ≈	5.0476e–1 –	1.0000e+0 ≈	2.5604e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW7	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW8	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.3333e–1 ≈	1.0000e+0
MW9	1.0000e+0 ≈	1.0000e+0 ≈	8.2418e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW10	1.0000e+0 ≈	1.0000e+0 ≈	9.7656e–1 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	9.6667e–1 ≈	1.0000e+0 ≈	5.0000e–1 –	1.0000e+0
MW11	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	7.1392e–1 –	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW12	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	8.6190e–1 –	1.0000e+0
MW13	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
MW14	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0 ≈	1.0000e+0
+/ – / ≈	0/0/38	0/6/32	0/3/35	0/9/29	0/1/37	0/1/37	0/0/38	0/0/38	0/7/31	

Table 7 The HV values of ten algorithms on 30 real-world CMOPs

Problem	CCMO	cDPEA	CMOEA_MS	DSPCMDE	TSTI	MTCMO	CMOEMT	TriP	C3M	TSRSM
RWMOP1	6.0667e–1 ≈	6.0662e–1 ≈	**7.6110e–1** +	9.3427e–2 –	6.0632e–1 ≈	6.0645e–1 ≈	6.0614e–1 –	6.0516e–1 –	6.0761e–1 +	6.0650e–1
RWMOP2	3.3447e–1 –	3.1090e–1 ≈	2.9907e–1 –	**3.9994e–1** +	2.3179e–1 –	3.9279-1 ≈	3.9288e–1 ≈	3.9268e–1 –	3.9285e–1 –	3.9288e–1
RWMOP3	**8.9984e–1** +	8.9891e–1 ≈	8.6287e–1 –	8.9494e–2 –	8.9863e–1 ≈	8.9891e–1 ≈	8.9784-1 –	8.9850e–1 ≈	8.9912e–1 ≈	8.9874e–1
RWMOP4	8.5812e–1 –	8.5839e–1 –	7.9822e–1 –	1.2582e–1 –	8.5767e–1 –	8.5865e–1 –	8.5551e–1 –	8.5986e–1 ≈	8.4675e–1 –	**8.6001e–1**
RWMOP5	**4.3479e–1** +	4.3452e–1 ≈	4.1973e–1 –	4.3275e–1 –	4.3441e–1 –	4.3469-1 +	4.3414e–1 –	4.3443e–1 –	4.1330e–1 –	4.3458e–1
RWMOP6	2.7699e–1 +	2.7687e–1 –	**2.7729e–1** +	2.7640e–1 –	2.7691e–1 ≈	2.7694e–1 ≈	2.7629e–1 –	2.7683e–1 –	2.7606e–1 –	2.7692e–1
RWMOP7	4.8443e–1 –	4.8443e–1 –	4.8401e–1 –	4.7819e–1 –	4.8443e–1 –	4.8442e–1 –	4.8450-1 ≈	4.8432e–1 –	**4.8461e–1** +	4.8449e–1
RWMOP8	2.6030e–2 ≈	2.5935e–2 –	**2.6164e–2** +	1.0233e–2 –	2.5415e–2 –	2.5961e–2 –	2.5893e–2 –	2.6006e–2 ≈	2.3548e–2 –	2.6004e–2
RWMOP9	4.0929e–1 –	4.0910e–1 –	4.0813e–1 –	5.2956e–2 –	4.0915e–1 –	4.0907e–1 –	4.0922e–1 –	4.0910e–1 –	4.0926e–1 –	**4.0948e–1**
RWMOP10	8.4125e–1 ≈	8.4036e–1 ≈	8.3856e–1 –	7.8343e–2 –	8.4127e–1 ≈	8.4034-1 ≈	8.4109e–1 ≈	**8.4190e–1** +	8.4126e–1 ≈	8.4108e–1
RWMOP11	9.0577e–2 –	9.4414e–2 ≈	8.8003e–2 –	7.2139e–2 –	**9.4479e–2** +	9.4074e–2 ≈	9.0345-2 –	9.3552e–2 ≈	7.9236e–2 –	9.3907e–2
RWMOP12	5.5379-1 ≈	5.5429-1 ≈	4.6705e–1 –	6.6866-2 –	5.5332e–1 ≈	5.5453e–1 ≈	5.5450e–1 ≈	5.5523e–1 ≈	**5.5533e–1** +	5.5439e–1
RWMOP13	8.7449e–2 –	8.7540e–2 –	8.1947e–2 –	8.7608e–2 ≈	8.7537e–2 –	8.7530e–2 ≈	8.6949e–2 –	8.7427e–2 –	7.6656e–2 –	**8.7620-2**
RWMOP14	**6.1721-1** +	6.1717e–1 +	3.7356e–1 –	4.4054e–1 –	6.1667e–1 ≈	6.1658e–1 ≈	6.1622e–1 –	6.1681e–1 ≈	5.7979e–1 –	6.1689e–1
RWMOP15	5.4336e–1 –	5.4253e–1 –	5.4330e–1 –	2.7586e–1 –	5.4320e–1 –	5.4354-1 +	5.4333e–1 –	5.4266e–1 –	**5.4361-1** +	5.4344e–1
RWMOP16	7.6219e–1 ≈	7.6213e–1 –	7.5403e–1 –	7.9630e–2 –	7.6221e–1 ≈	7.6229-1 ≈	7.6201e–1 –	7.6191e–1 –	7.6196e–1 –	**7.6231e–1**
RWMOP17	2.8280e–1 +	2.3112e–1 ≈	2.3018e–1 ≈	**4.0901e–1** +	3.4820e–1 +	4.0509-1 +	2.3421e–1 ≈	2.0261e–1 ≈	3.2575e–1 +	2.2109e–1
RWMOP18	4.0509-2 –	4.0509e–2 –	4.0512e–2 –	4.0343e–2 –	4.0509-2 –	4.0508e–2 –	**4.0515-2** ≈	4.0512-2 –	4.0504e–2 –	4.0514e–2
RWMOP19	3.4228e–1 –	2.9851e–1 –	3.3686e–1 –	2.6340-1 –	3.5503e–1 –	3.5779-1 –	3.6721e–1 –	3.4087e–1 –	NaN	**3.7089e–1**
RWMOP20	**0.0000e+0** ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0 ≈	0.0000e+0
RWMOP21	3.1756-2 –	3.1754e–2 –	3.1755e–2 –	2.9273e–2 –	3.1754e–2 –	3.1755e–2 –	3.1756-2 –	3.1755e–2 –	**3.1759e–2** +	3.1758-2
RWMOP22	5.4684e–1 ≈	5.6486-1 ≈	NaN	7.6818e–1 +	**1.4094e+0** ≈	5.4545e–1 ≈	7.3199e–1 ≈	6.6313e–1 ≈	7.7228e–1 ≈	6.6721e–1
RWMOP23	3.8043e–1 ≈	7.1260e–1 +	3.1004e–1 ≈	9.6212e–1 +	4.5863e–1 ≈	3.0020e–1 –	8.6661e–1 +	8.2595e–1 +	**9.9797e–1** +	4.4001e–1
RWMOP24	NaN	NaN	1.0000e+0 ≈	NaN	NaN	NaN	1.0000e+0 ≈	8.6813e–1 –	1.0000e+0≈	**1.0000e+0**
RWMOP25	2.4138e–1 –	2.4139e–1 ≈	2.3147e–1 –	2.3921e–1 –	2.3149e–1 –	2.4139e–1 –	2.4137e–1 –	2.4138-1 –	2.4138e–1 –	**2.4140-1**
RWMOP26	**1.4798e–1** ≈	1.3533e–1 –	1.3307e–1 –	1.4525e–1 –	1.2947e–1 –	1.4582e–1 –	1.4624e–1 ≈	1.4592e–1 –	1.4639e–1 ≈	1.4630e–1
RWMOP27	5.1256e+10 ≈	3.7824e+11 –	2.8901e+11 ≈	2.3266e+3 –	1.6198e+11 ≈	3.3086e+11 ≈	1.1699e+12 ≈	1.9674e+10 –	9.5642e+10 –	**1.8186e+12**
RWMOP28	5.2277e–3 ≈	1.0845e–2 ≈	1.9515e–2 ≈	2.1425e–2 ≈	2.4279e–2 +	**3.7608e–2** +	1.6653e–2 ≈	1.7912e–2 ≈	2.5448e–2 +	1.3817e–2
RWMOP29	7.7108e–1 –	NaN	7.8272e–1 –	NaN	7.8383e–1 –	7.8574-1 –	7.8673e–1 ≈	7.8427e–1 –	4.1264-1 –	**7.8694e–1**
RWMOP30	5.4444-1 ≈	NaN	NaN	NaN	5.8862-1 ≈	5.6446-1 ≈	5.5414-1 ≈	5.8374e–1 ≈	NaN	**6.1067-1**
+/ – / ≈	5/12/12	2/13/12	3/19/6	4/20/3	3/14/12	4/11/14	1/15/14	2/17/11	8/15/5	

Table 8 The IGD+ values of three algorithms with different switching mechanisms on three sets of benchmark test functions

Problem	TSRSM1	TSRSM2	TSRSM
C1_DTLZ1	1.4705e–2 (9.28-5) ≈	1.5291e–2 (3.75e–4) −	**1.4700e–2 (6.54-5)**
C1_DTLZ3	2.3970e–2 (1.01e–3) ≈	2.9286e–2 (1.63e–2) −	**2.3835e–2 (1.28-3)**
C2_DTLZ2	**1.9233e–2 (5.47e–4)** ≈	1.9254e–2 (6.32e–4) ≈	1.9268e–2 (6.03e–4)
C3_DTLZ4	5.862-2 (1.77e–3) ≈	5.9072e–2 (1.84e–3) ≈	**5.8194-2 (2.25e–3)**
DC1_DTLZ1	8.4431-3 (5.72e–5) ≈	**8.4410e–3 (6.96e–5)** ≈	8.4602e–3 (9.26-5)
DC1_DTLZ3	**1.3466e–2 (5.29e–4)** +	1.4133-2 (1.21e–3) ≈	1.7325e–2 (1.70e–2)
DC2_DTLZ1	**1.4701-2 (7.14e–5)** ≈	1.4820e–2 (4.72e–4) ≈	1.4953-2 (7.14e–4)
DC2_DTLZ3	**2.3539e–2 (8.00e–4)** ≈	1.3580e–1 (2.24e–1) ≈	1.0198-1 (1.95e–1)
DC3_DTLZ1	**5.1006e–3 (7.69-5)** ≈	2.2697e–2 (7.14e–2) ≈	1.0792-2 (3.00e–2)
DC3_DTLZ3	**4.4970e–2 (1.19e–1)** ≈	1.1806-1 (2.66-1) ≈	1.7186e–1 (2.60e–1)
LIRCMOP1	2.5627e–2 (9.78e–3) −	2.1819e–2 (8.49-3) −	**1.4963e–2 (8.07e–3)**
LIRCMOP2	2.0655-2 (8.35e–3) −	1.5877e–2 (5.62e–3) −	**9.0102e–3 (4.33e–3)**
LIRCMOP3	3.1131e–2 (1.79-2) −	2.2069e–2 (8.92e–3) −	**1.4317e–2 (8.99-3)**
LIRCMOP4	3.5835e–2 (2.77e–2) −	2.2270e–2 (1.34e–2) −	**1.5124-2 (1.12e–2)**
LIRCMOP5	**4.8785e–3 (3.70e–4)** +	5.0851e–3 (6.20e–3) ≈	5.6297-3 (1.02e–2)
LIRCMOP6	3.9614e–3 (1.88e–4) −	3.4128e–3 (3.5-4) ≈	**3.3155e–3 (3.19e–4)**
LIRCMOP7	6.6129e–3 (3.23e–4) ≈	6.5523-3 (3.53e–4) ≈	**6.5277e–3 (3.74e–4)**
LIRCMOP8	6.9915e–3 (3.41e–4) −	**6.3636-3 (3.73e–4)** ≈	6.4708e–3 (2.5-4)
LIRCMOP9	4.6677e–2 (2.53e–2) −	3.8850e–2 (2.51e–2) −	**2.2200e–2 (2.23-2)**
LIRCMOP10	4.5775e–3 (2.66e–4) −	4.5141e–3 (3.23-4) −	**4.1256e–3 (2.73e–4)**
LIRCMOP11	**7.1581-4 (3.11e–4)** ≈	8.9128e–4 (6.63e–4) ≈	2.3187-3 (9.08e–3)
LIRCMOP12	**5.2297e–4 (3.09e–4)** ≈	1.5648e–3 (4.27-3) ≈	7.9006e–4 (1.33e–3)
LIRCMOP13	4.2160e–2 (1.30e–3) ≈	**4.1835-2 (1.24e–3)** ≈	4.2186e–2 (1.16-3)
LIRCMOP14	4.9269e–2 (1.62e–3) −	4.8209e–2 (1.20e–3) ≈	**4.8073-2 (1.23e–3)**
MW1	**1.2486e–3 (1.13e–5)** ≈	1.2584-3 (4.42e–5) ≈	1.2487e–3 (1.08e–5)
MW2	**1.4773e–2 (7.95e–3)** ≈	1.6257e–2 (1.00-2) ≈	1.7687e–2 (7.60e–3)
MW3	**2.8256e–3 (1.88-4)** ≈	2.9552e–3 (2.26e–4) ≈	2.9021e–3 (2.79-4)
MW4	2.9880e–2 (2.67e–4) −	2.9780e–2 (1.96e–4) ≈	**2.9735e–2 (2.24e–4)**
MW5	**1.2946e–3 (1.75e–3)** ≈	1.8333e–3 (1.33e–3) +	6.6809e–3 (3.13e–2)
MW6	1.6796e–2 (1.20-2) ≈	**1.6420e–2 (9.89e–3)** ≈	1.6864e–2 (1.11e–2)
MW7	**2.0757-3 (1.23e–4)** ≈	2.1299e–3 (2.16e–4) ≈	2.0949e–3 (1.60-4)
MW8	2.9931e–2 (6.97e–3) ≈	2.7604-2 (5.46e–3) ≈	**2.7142e–2 (7.45e–3)**
MW9	2.4188e–2 (1.14-1) ≈	3.3943e–3 (2.74-4) ≈	**3.3726e–3 (2.62e–4)**
MW10	2.9364e–2 (3.15-2) ≈	2.7341e–2 (2.07e–2) ≈	**2.2101e–2 (1.71e–2)**
MW11	2.9415-3 (1.18e–4) ≈	**2.9241-3 (9.90e–5)** ≈	2.9800e–3 (1.60e–4)
MW12	**3.3533-3 (1.54e–4)** ≈	2.4905e–2 (1.18e–1) ≈	3.3639e–3 (1.75-4)
MW13	**2.3036e–2 (1.89e–2)** ≈	2.9542e–2 (1.70e–2) ≈	2.3501e–2 (1.21-2)
MW14	**6.4272e–2 (2.13e–3)** ≈	6.6568e–2 (3.40-3) −	6.4823-2 (2.19e–3)
+/ − / ≈	2/11/25	1/9/28	

Table 9 The HV values of three algorithms with different switching mechanisms on three sets of benchmark test functions

Problem	TSRSM1	TSRSM2	TSRSM
C1_DTLZ1	8.4137e–1 (1.93e–4) ≈	8.3889e–1 (1.50-3) −	**8.4144e–1 (1.37e–4)**
C1_DTLZ3	5.5756e–1 (1.09e–3) ≈	5.4917e–1 (2.67e–2) −	**5.5858-1 (2.12e–3)**
C2_DTLZ2	5.1736e–1 (1.79e–3) ≈	**5.1752-1 (9.91e–4)** ≈	5.1711e–1 (1.09-3)
C3_DTLZ4	7.8790e–1 (1.14e–3) ≈	7.8770e–1 (1.14-3) ≈	**7.8815-1 (1.49e–3)**
DC1_DTLZ1	**6.3240-1 (3.91e–4)** ≈	6.3226e–1 (6.14e–4) ≈	6.3224e–1 (6.44e–4)
DC1_DTLZ3	**4.7365e–1 (9.23e–4)** +	4.7178e–1 (2.51e–3) ≈	4.6630e–1 (3.04e–2)
DC2_DTLZ1	**8.4147e–1 (1.27e–4)** ≈	8.4121e–1 (1.11e–3) ≈	8.4093e–1 (1.58e–3)
DC2_DTLZ3	**5.5919e–1 (1.14e–3)** ≈	4.4601e–1 (2.25e–1) ≈	4.8035e–1 (1.96e–1)
DC3_DTLZ1	**5.3609e–1 (6.25e–4)** ≈	5.0409e–1 (1.16e–1) ≈	5.2057e–1 (7.96e–2)
DC3_DTLZ3	**3.3243e–1 (8.84e–2)** ≈	2.9338e–1 (1.28e–1) ≈	2.5323e–1 (1.59e–1)
LIRCMOP1	2.2147e–1 (8.05e–3) −	2.2544e–1 (6.29-3) −	**2.3071e–1 (5.53e–3)**
LIRCMOP2	3.4831e–1 (6.40e–3) −	3.5255e–1 (3.92-3) −	**3.5740e–1 (2.67e–3)**
LIRCMOP3	1.8971e–1 (8.02e–3) −	1.9486e–1 (6.63e–3) −	**1.9929e–1 (5.84e–3)**
LIRCMOP4	2.9309e–1 (1.65e–2) −	3.0151e–1 (9.88e–3) −	**3.0688-1 (8.24e–3)**
LIRCMOP5	**2.9197e–1 (1.66e–4)** +	2.9189e–1 (2.77e–3) ≈	2.9068e–1 (9.84e–3)
LIRCMOP6	1.9756e–1 (9.81e–5) −	1.9785e–1 (1.88e–4) ≈	**1.9790e–1 (1.68-4)**
LIRCMOP7	2.8418e–1 (1.36e–4) ≈	2.9414e–1 (1.43-4) ≈	**2.9415e–1 (1.63e–4)**
LIRCMOP8	2.8422e–1 (1.52e–4) −	**2.9423e–1 (1.51e–4)** ≈	2.9418e–1 (1.10e–4)
LIRCMOP9	5.4510e–1 (1.28e–2) −	5.4862e–1 (1.19e–2) −	**5.5690e–1 (9.27e–3)**
LIRCMOP10	7.0742e–1 (2.05e–4) −	7.0748e–1 (2.64e–4) −	**7.0776e–1 (1.73e–4)**
LIRCMOP11	**6.9388e–1 (2.59e–4)** ≈	6.9379e–1 (4.95e–4) ≈	6.9282e–1 (6.16-3)
LIRCMOP12	**6.2025e–1 (1.55-4)** ≈	6.1981e–1 (1.79e–3) ≈	6.2012-1 (6.82e–4)
LIRCMOP13	5.5622-1 (1.27-3) −	**5.5742e–1 (1.25-3)** ≈	5.5715e–1 (1.02e–3)
LIRCMOP14	5.5101-1 (1.49e–3) −	5.5169-1 (1.49e–3) ≈	**5.520-1 (1.24e–3)**
MW1	**4.9001-1 (2.09e–5)** ≈	4.8999-1 (7.25-5) ≈	4.9000e–1 (2.25-5)
MW2	**5.6185e–1 (1.31e–2)** ≈	5.5954e–1 (1.58e–2) ≈	5.5704-1 (1.21e–2)
MW3	**5.4457e–1 (3.24e–4)** ≈	5.4429-1 (3.87e–4) ≈	5.4442e–1 (4.68-4)
MW4	8.4097e–1 (2.85e–4) −	8.4103-1 (2.72e–4) ≈	**8.4110e–1 (3.16e–4)**
MW5	**3.2398e–1 (1.02e–3)** ≈	3.2378-1 (6.63e–4) +	3.1947e–1 (2.57e–2)
MW6	3.0695-1 (1.64e–2) ≈	**3.0747e–1 (1.34e–2)** ≈	3.0690-1 (1.51e–2)
MW7	4.1231e–1 (2.84e–4) ≈	4.1214e–1 (4.71-4) ≈	**4.1224e–1 (3.42e–4)**
MW8	5.3331-1 (1.26e–2) ≈	5.3735e–1 (1.01e–2) ≈	**5.3830-1 (1.35e–2)**
MW9	3.8450e–1 (7.26-2) ≈	**3.9771e–1 (2.14e–3)** ≈	3.9740e–1 (2.10-3)
MW10	4.2512e–1 (2.55e–2) ≈	4.2583-1 (1.86e–2) ≈	**4.3043e–1 (1.60e–2)**
MW11	4.4750-1 (2.01e–4) ≈	4.4754-1 (1.63e–4) ≈	**4.4758e–1 (1.59-4)**
MW12	6.0411e–1 (2.23e–4) ≈	5.8471e–1 (1.07-1) ≈	**6.0419e–1 (3.37e–4)**
MW13	**4.5937-1 (1.41e–2)** ≈	4.5407e–1 (1.32-2) ≈	4.5873e–1 (1.05e–2)
MW14	**4.7432-1 (1.14e–3)** ≈	4.730-1 (2.38-3) ≈	4.7407e–1 (1.45e–3)
+/ − / ≈	2/11/25	1/8/29	

Table 10 The IGD+ values of four algorithms with different stage order on three sets of benchmark test functions

Problem	TSRSM-PRP	TSRSM-PP	TSRSM-PR	TSRSM
C1_DTLZ1	1.5342e–2 (1.93-4) −	1.5644e–2 (3.52e–4) −	1.4714-2 (9.89e–5) ≈	**1.4700e–2 (6.54-5)**
C1_DTLZ3	2.4094e–2 (1.21e–3) ≈	2.6805e–2 (1.90-3) −	**2.3296e–2 (8.78e–4)** +	2.3835-2 (1.28e–3)
C2_DTLZ2	2.0242e–2 (5.32e–4) −	1.9575-2 (5.51e–4) ≈	**1.8989e–2 (3.67-4)** +	1.9268e–2 (6.03e–4)
C3_DTLZ4	6.4276e–2 (2.63e–3) −	6.1454-2 (2.52e–3) −	5.8947e–2 (1.95e–3) ≈	**5.8194e–2 (2.25e–3)**
DC1_DTLZ1	9.9045e–3 (3.66e–4) −	8.9130e–3 (1.51e–4) −	8.5267e–3 (9.29e–5) −	**8.4602e–3 (9.26e–5)**
DC1_DTLZ3	1.5569e–2 (9.87e–4) +	1.5768e–2 (1.35e–3) +	**1.3430e–2 (4.02e–4)** +	1.7325e–2 (1.70e–2)
DC2_DTLZ1	1.5374e–2 (1.96e–4) −	2.5945e–2 (3.86e–2) −	**1.4740e–2 (1.24e–4)** ≈	1.4953e–2 (7.14e–4)
DC2_DTLZ3	7.9450e–2 (1.66e–1) +	3.8650e–1 (2.62e–1) −	**2.3337e–2 (9.59e–4)** ≈	1.0198e–1 (1.95e–1)
DC3_DTLZ1	5.7257e–3 (3.52e–4) +	3.4267e–2 (6.55e–2) −	**5.1266e–3 (1.24e–4)** ≈	1.0792e–2 (3.00e–2)
DC3_DTLZ3	4.5723e–2 (1.39e–1) +	8.4798e–1 (4.16e–1) −	**8.2606e–3 (4.65e–4)** +	1.7186e–1 (2.60e–1)
LIRCMOP1	7.3836e–2 (5.15e–2) −	**1.1898e–2 (4.45e–3)** ≈	1.3985e–1 (5.27e–2) −	1.4963e–2 (8.07e–3)
LIRCMOP2	4.6837e–2 (2.00e–2) −	**8.0214e–3 (3.77e–3)** ≈	6.2454e–2 (2.86e–2) −	9.0102e–3 (4.33e–3)
LIRCMOP3	1.5829e–1 (4.22e–2) −	1.4884e–2 (8.09e–3) ≈	1.4886e–1 (4.31e–2) −	**1.4317e–2 (8.99e–3)**
LIRCMOP4	1.2542e–1 (4.46e–2) −	**1.0854e–2 (7.41e–3)** ≈	1.1530e–1 (3.91e–2) −	1.5124e–2 (1.12e–2)
LIRCMOP5	1.0046e–2 (7.55e–3) −	**3.2027e–3 (8.08e–5)** +	1.2374e–2 (4.91e–3) −	5.6297e–3 (1.02e–2)
LIRCMOP6	5.9529e–3 (2.97e–3) −	**2.7760e–3 (4.88e–5)** +	1.2702e–2 (7.30e–3) −	3.3155e–3 (3.19e–4)
LIRCMOP7	7.2516e–3 (9.79e–4) −	**6.2496e–3 (3.25e–4)** +	8.5088e–3 (2.87e–3) −	6.5277e–3 (3.74e–4)
LIRCMOP8	6.5356e–3 (1.41e–3) −	**5.5745e–3 (1.85e–4)** +	6.7493e–3 (1.68e–3) −	6.4708e–3 (2.57e–4)
LIRCMOP9	3.4279e–2 (2.80e–2) −	2.4655e–2 (3.05e–2) ≈	**2.1353e–2 (1.95e–2)** ≈	2.2200e–2 (2.23e–2)
LIRCMOP10	3.4298e–3 (3.93e–4) +	**3.1127e–3 (1.50e–4)** +	5.1280e–3 (3.94e–4) −	4.1256e–3 (2.73e–4)
LIRCMOP11	7.6258e–4 (1.32e–4) +	4.3620e–3 (1.26e–2) ≈	**6.7081e–4 (1.03e–4)** ≈	2.3187e–3 (9.08e–3)
LIRCMOP12	7.9841e–4 (9.59e–4) −	**4.4733e–4 (6.81e–5)** ≈	1.4517e–3 (1.98e–3) −	7.9006e–4 (1.33e–3)
LIRCMOP13	4.4937e–2 (1.49e–3) −	4.4847e–2 (1.06e–3) −	4.2212e–2 (1.23e–3) ≈	**4.2186e–2 (1.16e–3)**
LIRCMOP14	5.0733e–2 (1.29e–3) −	**4.7243e–2 (1.36e–3)** +	4.7933e–2 (1.69e–3) ≈	4.8073e–2 (1.23e–3)
MW1	1.4868e–3 (5.82e–5) −	1.3728e–3 (4.14e–5) −	1.2588e–3 (2.84e–5) ≈	**1.2487e–3 (1.08e–5)**
MW2	1.5326e–2 (6.84e–3) ≈	**1.4827e–2 (9.38e–3)** ≈	1.5316e–2 (7.02e–3) ≈	1.7687e–2 (7.60e–3)
MW3	**2.7785e–3 (2.92e–4)** +	2.7994e–3 (2.85e–4) ≈	2.9155e–3 (3.04e–4) ≈	2.9021e–3 (2.79e–4)
MW4	3.3068e–2 (1.03e–3) −	3.2757e–2 (8.58e–4) −	2.9766e–2 (2.88e–4) ≈	**2.9735e–2 (2.24e–4)**
MW5	2.0588e–3 (2.00e–3) +	**1.1336e–3 (8.35e–4)** ≈	1.7987e–3 (2.64e–3) ≈	6.6809e–3 (3.13e–2)
MW6	1.5487e–2 (1.11e–2) ≈	1.4776e–2 (8.48e–3) ≈	**1.3517e–2 (8.69e–3)** ≈	1.6864e–2 (1.11e–2)
MW7	2.3707e–3 (3.45e–4) −	2.3914e–3 (2.36e–4) −	**2.0653e–3 (1.62e–4)** ≈	2.0949e–3 (1.60e–4)
MW8	3.0494e–2 (7.98e–3) −	2.8186e–2 (6.18e–3) ≈	2.9529e–2 (9.23e–3) ≈	**2.7142e–2 (7.45e–3)**
MW9	3.7062e–3 (4.30e–4) −	5.5843e–3 (1.79e–3) −	**3.2019e–3 (1.33e–4)** +	3.3726e–3 (2.62e–4)
MW10	2.5388e–2 (2.17e–2) ≈	**2.1123e–2 (1.51e–2)** ≈	2.1243e–2 (1.58e–2) ≈	2.2101e–2 (1.71e–2)
MW11	3.3886e–3 (2.12e–4) −	3.3978e–3 (1.74e–4) −	**2.8968e–3 (1.77e–4)** ≈	2.9800e–3 (1.60e–4)
MW12	3.8221e–3 (1.63e–4) −	3.6857e–3 (1.44e–4) −	2.7719e–2 (1.34e–1) −	**3.3639e–3 (1.75e–4)**
MW13	3.3163e–2 (1.65e–2) −	3.2624e–2 (2.28e–2) ≈	3.1377e–2 (1.57e–2) ≈	**2.3501e–2 (1.21e–2)**
MW14	7.1087e–2 (2.64e–3) −	7.2179e–2 (2.64e–3) −	**6.3440e–2 (1.72e–3)** +	6.4823e–2 (2.19e–3)
+/ − / ≈	8/26/4	7/16/15	6/12/20	

Table 11 The HV values of four algorithms with different stage order on three sets of benchmark test functions

Problem	TSRSM-PRP	TSRSM-PP	TSRSM-PR	TSRSM
C1_DTLZ1	8.3722e–1 (1.85e–3) −	8.3778e–1 (9.80e–4) −	8.3987e–1 (2.42e–3) −	**8.4144e–1 (1.37e–4)**
C1_DTLZ3	5.5781e–1 (2.15e–3) ≈	5.5313e–1 (2.98e–3) −	**5.5949e–1 (1.37e–3)** +	5.5858e–1 (2.12e–3)
C2_DTLZ2	5.1299e–1 (1.63e–3) −	5.1522e–1 (1.47e–3) −	**5.1739e–1 (1.07e–3)** ≈	5.1711e–1 (1.09e–3)
C3_DTLZ4	7.8429e–1 (1.66e–3) −	7.8599e–1 (1.58e–3) −	7.8779e–1 (1.22e–3) ≈	**7.8815e–1 (1.49e–3)**
DC1_DTLZ1	6.2692e–1 (1.51e–3) −	6.2955e–1 (8.17e–4) −	6.3202e–1 (6.00e–4) −	**6.3224e–1 (6.44e–4)**
DC1_DTLZ3	4.6945e–1 (1.63e–3) +	4.6826e–1 (2.50e–3) +	**4.7366e–1 (9.87e–4)** +	4.6630e–1 (3.04e–2)
DC2_DTLZ1	8.3979e–1 (4.00e–4) −	8.1381e–1 (9.57e–2) −	**8.4135e–1 (2.37e–4)** ≈	8.4093e–1 (1.58e–3)
DC2_DTLZ3	5.0165e–1 (1.66e–1) +	1.9350e–1 (2.62e–1) −	**5.5968e–1 (1.48e–3)** ≈	4.8035e–1 (1.96e–1)
DC3_DTLZ1	5.3273e–1 (1.54e–3) +	4.5770e–1 (1.73e–1) −	**5.3555e–1 (1.30e–3)** ≈	5.2057e–1 (7.96e–2)
DC3_DTLZ3	3.4111e–1 (9.27e–2) +	0.0000e+0 (0.00e+0) −	**3.6780e–1 (8.96e–4)** +	2.5323e–1 (1.59e–1)
LIRCMOP1	1.9268e–1 (2.55e–2) −	**2.3285e–1 (2.79e–3)** ≈	1.5983e–1 (1.85e–2) −	2.3071e–1 (5.53e–3)
LIRCMOP2	3.2020e–1 (1.93e–2) −	**3.5801e–1 (2.22e–3)** ≈	3.0588e–1 (2.53e–2) −	3.5740e–1 (2.67e–3)
LIRCMOP3	1.3547e–1 (1.66e–2) −	**1.9986e–1 (4.85e–3)** ≈	1.3902e–1 (1.53e–2) −	1.9929e–1 (5.84e–3)
LIRCMOP4	2.2917e–1 (2.75e–2) −	**3.0992e–1 (5.34e–3)** ≈	2.3508e–1 (2.60e–2) −	3.0688e–1 (8.24e–3)
LIRCMOP5	2.8549e–1 (9.77e–3) −	**2.9277e–1 (4.93e–5)** +	2.8539e–1 (6.34e–3) −	2.9068e–1 (9.84e–3)
LIRCMOP6	1.9630e–1 (1.88e–3) −	**1.9818e–1 (3.17e–5)** +	1.9242e–1 (3.47e–3) −	1.9790e–1 (1.68e–4)
LIRCMOP7	2.9349e–1 (1.18e–3) −	**2.9428e–1 (1.36e–4)** +	2.9223e–1 (2.32e–3) −	2.9415e–1 (1.63e–4)
LIRCMOP8	2.9401e–1 (1.44e–3) −	**2.9460e–1 (7.93e–5)** +	2.9351e–1 (1.94e–3) −	2.9418e–1 (1.10e–4)
LIRCMOP9	5.5107e–1 (1.32e–2) −	5.5550e–1 (1.52e–2) ≈	5.5640e–1 (8.50e–3) ≈	**5.5690e–1 (9.27e–3)**
LIRCMOP10	7.0777e–1 (3.07e–4) ≈	**7.0826e–1 (1.06e–4)** +	7.0660e–1 (3.86e–4) −	7.0776e–1 (1.73e–4)
LIRCMOP11	6.9387e–1 (1.01e–4) +	6.9144e–1 (8.57e–3) ≈	**6.9390e–1 (1.51e–4)** ≈	6.9282e–1 (6.16e–3)
LIRCMOP12	6.2011e–1 (5.00e–4) −	**6.2030e–1 (2.26e–5)** +	6.1977e–1 (9.61e–4) −	6.2012e–1 (6.82e–4)
LIRCMOP13	5.5365e–1 (1.61e–3) −	5.5368e–1 (1.26e–3) −	5.5697e–1 (1.01e–3) ≈	**5.5715e–1 (1.02e–3)**
LIRCMOP14	5.4916e–1 (1.39e–3) −	**5.5261e–1 (1.17e–3)** ≈	5.5211e–1 (1.57e–3) ≈	5.5201e–1 (1.24e–3)
MW1	4.8977e–1 (7.16e–5) −	4.8979e–1 (6.32e–5) −	4.8998e–1 (6.60e–5) ≈	**4.9000e–1 (2.25e–5)**
MW2	5.6062e–1 (1.13e–2) ≈	**5.6199e–1 (1.53e–2)** ≈	5.6059e–1 (1.16e–2) ≈	5.5704e–1 (1.21e–2)
MW3	**5.4456e–1 (5.03e–4)** ≈	5.4451e–1 (5.20e–4) ≈	5.4437e–1 (4.95e–4) ≈	5.4442e–1 (4.68e–4)
MW4	8.3694e–1 (1.19e–3) −	8.3731e–1 (8.77e–4) −	8.4110e–1 (2.74e–4) ≈	**8.4110e–1 (3.16e–4)**
MW5	3.2361e–1 (1.04e–3) +	**3.2404e–1 (4.19e–4)** ≈	3.2367e–1 (1.98e–3) ≈	3.1947e–1 (2.57e–2)
MW6	3.0877e–1 (1.51e–2) ≈	3.0964e–1 (1.15e–2) ≈	**3.1140e–1 (1.18e–2)** ≈	3.0690e–1 (1.51e–2)
MW7	4.1188e–1 (5.92e–4) −	4.1199e–1 (4.47e–4) ≈	**4.1232e–1 (3.00e–4)** ≈	4.1224e–1 (3.42e–4)
MW8	5.3061e–1 (1.45e–2) −	5.3504e–1 (1.10e–2) −	5.3388e–1 (1.63e–2) ≈	**5.3830e–1 (1.35e–2)**
MW9	3.9582e–1 (2.66e–3) −	3.9113e–1 (4.05e–3) −	**3.9857e–1 (1.40e–3)** +	3.9740e–1 (2.10e–3)
MW10	4.2809e–1 (1.99e–2) ≈	4.3123e–1 (1.47e–2) ≈	**4.3151e–1 (1.61e–2)** ≈	4.3043e–1 (1.60e–2)
MW11	4.4736e–1 (2.13e–4) −	4.4756e–1 (1.23e–4) ≈	4.4731e–1 (2.99e–4) −	**4.4758e–1 (1.59e–4)**
MW12	6.0354e–1 (2.86e–4) −	6.0366e–1 (2.76e–4) −	5.8425e–1 (1.10e–1) −	**6.0419e–1 (3.37e–4)**
MW13	4.5100e–1 (1.20e–2) −	4.5194e–1 (1.63e–2) ≈	4.5242e–1 (1.20e–2) ≈	**4.5873e–1 (1.05e–2)**
MW14	4.6910e–1 (1.88e–3) −	4.6889e–1 (2.41e–3) −	**4.7478e–1 (1.12e–3)** +	4.7407e–1 (1.45e–3)
+/ − / ≈	6/26/6	7/16/15	5/14/19	

Table 12 The IGD+ values of TSRSM and three variants with different strategies on LIRCMOP test suits

Problem	TSRAM-V1	TSRSM-V2	TSRSM-V3	TSRSM-V4	TSRSM
LIRCMOP1	**1.1807e–2 (4.74e–3)** ≈	1.3744e–1 (5.80e–2) –	1.6252e–2 (7.56e–3) ≈	5.1092e–2 (3.15e–2) –	1.4963e–2 (8.07e–3)
LIRCMOP2	**7.8186e–3 (2.40e–3)** ≈	6.7025e–2 (2.67e–2) –	1.8050e–2 (1.93e–2) –	4.9646e–2 (2.22e–2) –	9.0102e–3 (4.33e–3)
LIRCMOP3	1.6932e–2 (1.88e–2) ≈	1.5909e–1 (5.49e–2) –	4.0328e–2 (5.51e–2) –	7.0642e–2 (4.42e–2) –	**1.4317e–2 (8.99e–3)**
LIRCMOP4	**1.0239e–2 (7.01e–3)** ≈	1.2211e–1 (3.15e–2) –	2.7099e–2 (2.35e–2) –	6.3266e–2 (2.68e–2) –	1.5124e–2 (1.12e–2)
LIRCMOP5	**4.2601e–3 (3.38e–4)** +	7.4137e–3 (1.92e–3) –	1.1294e–2 (9.91e–3) –	1.1148e–2 (1.65e–2) –	5.6297e–3 (1.02e–2)
LIRCMOP6	3.7469e–3 (1.48e–3) –	7.1545e–3 (3.37e–3) –	1.1824e–2 (7.87e–3) –	8.1168e–3 (1.17e–2) –	**3.3155e–3 (3.19e–4)**
LIRCMOP7	8.5319e–3 (4.25e–3) –	7.2269e–3 (1.18e–3) –	9.1405e–3 (5.50e–3) –	7.3307e–3 (1.64e–3) –	**6.5277e–3 (3.74e–4)**
LIRCMOP8	6.6043e–3 (4.36e–4) –	6.5011e–3 (3.07e–4) ≈	7.2342e–3 (1.81e–3) ≈	6.7375e–3 (4.41e–4) –	**6.4708e–3 (2.57e–4)**
LIRCMOP9	2.7543e–2 (2.95e–2) –	3.9759e–2 (2.12e–2) –	3.9965e–2 (2.64e–2) –	4.2223e–2 (2.73e–2) –	**2.2200e–2 (2.23e–2)**
LIRCMOP10	4.1852e–3 (3.82e–4) ≈	5.3266e–3 (5.21e–4) –	4.9569e–3 (1.16e–3) –	4.6624e–3 (2.42e–4) –	**4.1256e–3 (2.73e–4)**
LIRCMOP11	7.9066e–4 (6.39e–4) ≈	7.3341e–4 (2.24e–4) ≈	**7.0788e–4 (7.96e–5)** ≈	1.3289e–3 (3.19e–3) +	2.3187e–3 (9.08e–3)
LIRCMOP12	**4.6233e–4 (9.79e–5)** ≈	1.0229e–3 (9.85e–4) ≈	7.8062e–4 (1.49e–3) ≈	7.9628e–3 (1.59e–2) –	7.9006e–4 (1.33e–3)
LIRCMOP13	4.2546e–2 (1.13e–3) ≈	4.2124e–2 (1.34e–3) ≈	**4.2073e–2 (1.31e–3)** ≈	4.2115e–2 (1.44e–3) ≈	4.2186e–2 (1.16e–3)
LIRCMOP14	4.8095e–2 (1.36e–3) ≈	4.8261e–2 (1.22e–3) ≈	4.8438e–2 (1.28e–3) ≈	4.8272e–2 (1.45e–3) ≈	**4.8073e–2 (1.23e–3)**
+/ – / ≈	1/4/9	0/9/5	0/8/6	1/11/2	

Table 13 The IGD+ values of five algorithms with different gr_1 on three sets of benchmark test functions

Problem	TSRSM3	TSRSM4	TSRSM5	TSRSM6	TSRSM
C1_DTLZ1	1.4686e–2 (9.20e–5) ≈	**1.4681e–2 (1.06e–4)** ≈	1.4682e–2 (6.88e–5) ≈	1.4690e–2 (9.21e–5) ≈	1.4700e–2 (6.54e–5)
C1_DTLZ3	3.0529e–2 (2.16e–2) −	2.4356e–2 (1.10e–3) −	2.4897e–2 (4.05e–3) ≈	2.4137e–2 (1.34e–3) ≈	**2.3835e–2 (1.28e–3)**
C2_DTLZ2	1.9166e–2 (5.51e–4) ≈	1.9325e–2 (7.87e–4) ≈	**1.9077e–2 (6.49e–4)** ≈	1.9206e–2 (5.54e–4) ≈	1.9268e–2 (6.03e–4)
C3_DTLZ4	5.9083e–2 (1.87e–3) ≈	5.8656e–2 (2.21e–3) ≈	5.8571e–2 (1.99e–3) ≈	5.8770e–2 (2.32e–3) ≈	**5.8194e–2 (2.25e–3)**
DC1_DTLZ1	8.4975e–3 (2.12e–4) ≈	**8.4511e–3 (6.14e–5)** ≈	8.4537e–3 (6.21e–5) ≈	8.4543e–3 (5.19e–5) ≈	8.4602e–3 (9.26e–5)
DC1_DTLZ3	2.4165e+0 (2.33e+0) −	1.1855e+0 (5.04e–1) −	1.0670e+0 (3.80e–1) −	1.3295e+0 (4.65e–1) −	**1.7325e–2 (1.70e–2)**
DC2_DTLZ1	2.1128e–2 (2.82e–2) −	1.4710e–2 (7.99e–5) ≈	1.4687e–2 (7.34e–5) +	**1.4684e–2 (6.46e–5)** +	1.4953e–2 (7.14e–4)
DC2_DTLZ3	1.2173e–1 (2.03e–1) ≈	7.9025e–2 (1.68e–1) ≈	**2.4005e–2 (1.23e–3)** ≈	4.1984e–2 (9.94e–2) ≈	1.0198e–1 ()
DC3_DTLZ1	7.9788e–2 (1.99e–1) −	5.2936e–3 (5.22e–4) ≈	5.1951e–3 (7.22e–4) +	**5.0788e–3 (6.07e–5)** ≈	1.0792e–2 (3.00e–2)
DC3_DTLZ3	1.0915e+0 (1.25e+0) −	1.5575e–1 (2.94e–1) ≈	**1.5138e–2 (2.01e–2)** +	6.4118e–2 (1.67e–1) +	1.7186e–1 (2.60e–1)
LIRCMOP1	1.6711e–2 (8.38e–3) ≈	1.6237e–2 (8.40e–3) ≈	**1.4190e–2 (5.16e–3)** ≈	1.5249e–2 (5.47e–3) ≈	1.4963e–2 (8.07e–3)
LIRCMOP2	1.3406e–2 (9.16e–3) −	**8.9265e–3 (3.11e–3)** ≈	9.1490e–3 (3.30e–3) ≈	1.0316e–2 (3.84e–3) −	9.0102e–3 (4.33e–3)
LIRCMOP3	1.8970e–2 (1.27e–2) ≈	1.6788e–2 (9.47e–3) ≈	1.9905e–2 (1.13e–2) ≈	2.0691e–2 (1.27e–2) −	**1.4317e–2 (8.99e–3)**
LIRCMOP4	1.5705e–2 (1.25e–2) ≈	1.5813e–2 (9.96e–3) ≈	1.7045e–2 (1.19e–2) ≈	2.2481e–2 (2.54e–2) ≈	**1.5124e–2 (1.12e–2)**
LIRCMOP5	1.9102e–2 (6.92e–2) −	**3.8316e–3 (5.08e–4)** ≈	4.8536e–3 (4.09e–3) ≈	5.1596e–3 (7.76e–3) ≈	5.6297e–3 (1.02e–2)
LIRCMOP6	3.5135e–2 (6.33e–2) −	**3.1293e–3 (2.73e–4)** +	3.1901e–3 (3.39e–4) ≈	3.9350e–3 (2.13e–3) ≈	3.3155e–3 (3.19e–4)
LIRCMOP7	8.5948e–3 (1.17e–2) ≈	6.6754e–3 (3.41e–4) ≈	6.5972e–3 (4.92e–4) ≈	6.7716e–3 (9.06e–4) ≈	**6.5277e–3 (3.74e–4)**
LIRCMOP8	6.5176e–3 (9.83e–4) ≈	6.4742e–3 (5.56e–4) ≈	**6.3780e–3 (3.22e–4)** ≈	6.6853e–3 (5.91e–4) ≈	6.4708e–3 (2.57e–4)
LIRCMOP9	7.1623e–2 (1.76e–2) −	3.3845e–2 (2.33e–2) ≈	3.8794e–2 (2.42e–2) −	4.2616e–2 (2.82e–2) −	**2.2200e–2 (2.23e–2)**
LIRCMOP10	4.9867e–3 (1.62e–3) −	4.2410e–3 (2.72e–4) ≈	4.2533e–3 (3.43e–4) ≈	4.2752e–3 (2.67e–4) −	**4.1256e–3 (2.73e–4)**
LIRCMOP11	1.5893e–2 (2.53e–2) −	**6.7752e–4 (1.16e–4)** ≈	1.5102e–3 (4.18e–3) ≈	1.4569e–3 (4.24e–3) ≈	2.3187e–3 (9.08e–3)
LIRCMOP12	2.4283e–2 (2.16e–2) −	8.4884e–4 (1.54e–3) ≈	**4.6502e–4 (1.47e–4)** ≈	1.2251e–3 (2.99e–3) ≈	7.9006e–4 (1.33e–3)
LIRCMOP13	4.2385e–2 (1.27e–3) ≈	4.2360e–2 (1.22e–3) ≈	4.2480e–2 (9.37e–4) ≈	4.2191e–2 (8.78e–4) ≈	**4.2186e–2 (1.16e–3)**
LIRCMOP14	**4.8049e–2 (1.23e–3)** ≈	4.8336e–2 (1.41e–3) ≈	4.8315e–2 (1.33e–3) ≈	4.8205e–2 (1.71e–3) ≈	4.8073e–2 (1.23e–3)
MW1	3.2268e–2 (1.38e–1) −	1.3766e–3 (6.56e–4) ≈	1.2491e–3 (1.06e–5) ≈	1.2496e–3 (1.37e–5) ≈	**1.2487e–3 (1.08e–5)**
MW2	7.7915e–2 (7.55e–2) −	1.5155e–2 (7.27e–3) ≈	**1.4459e–2 (7.40e–3)** ≈	1.5457e–2 (8.36e–3) ≈	1.7687e–2 (7.60e–3)
MW3	2.9942e–3 (2.54e–4) ≈	2.8788e–3 (2.66e–4) ≈	2.8187e–3 (2.22e–4) ≈	**2.7991e–3 (1.84e–4)** ≈	2.9021e–3 (2.79e–4)
MW4	3.1517e–2 (3.08e–3) −	3.0199e–2 (2.20e–3) ≈	2.9832e–2 (2.67e–4) ≈	2.9787e–2 (2.43e–4) ≈	**2.9735e–2 (2.24e–4)**
MW5	1.1996e–1 (1.50e–1) −	**9.4369e–4 (7.03e–4)** ≈	1.0398e–3 (1.34e–3) ≈	4.9559e–3 (1.80e–2) ≈	6.6809e–3 (3.13e–2)
MW6	2.1318e–1 (2.77e–1) −	1.6208e–2 (1.37e–2) ≈	1.6125e–2 (9.00e–3) ≈	**1.2218e–2 (7.46e–3)** ≈	1.6864e–2 (1.11e–2)
MW7	2.2564e–3 (2.57e–4) −	**2.0741e–3 (1.77e–4)** ≈	2.0787e–3 (2.26e–4) ≈	2.0766e–3 (1.57e–4) ≈	2.0949e–3 (1.60e–4)
MW8	7.0143e–2 (5.32e–2) −	**2.5153e–2 (4.90e–3)** ≈	2.6015e–2 (5.15e–3) ≈	2.7834e–2 (1.06e–2) ≈	2.7142e–2 (7.45e–3)
MW9	6.8779e–2 (1.92e–1) −	**3.2336e–3 (1.35e–4)** +	3.3258e–3 (2.29e–4) ≈	3.2627e–3 (1.60e–4) ≈	3.3726e–3 (2.62e–4)
MW10	2.6980e–1 (1.84e–1) −	2.3299e–2 (1.61e–2) ≈	3.9404e–2 (9.49e–2) ≈	2.4515e–2 (1.75e–2) ≈	**2.2101e–2 (1.71e–2)**
MW11	1.3764e–1 (2.30e–1) −	2.3369e–2 (6.96e–2) −	2.5797e–2 (6.87e–2) −	2.4338e–2 (4.74e–2) −	**2.9800e–3 (1.60e–4)**
MW12	6.2306e–2 (1.80e–1) −	3.3310e–3 (1.92e–4) ≈	3.3405e–3 (1.89e–4) ≈	**3.3230e–3 (1.44e–4)** ≈	3.3639e–3 (1.75e–4)
MW13	2.3545e–1 (1.99e–1) −	3.0468e–2 (2.12e–2) ≈	2.9195e–2 (2.65e–2) ≈	2.6802e–2 (1.61e–2) ≈	**2.3501e–2 (1.21e–2)**
MW14	6.7370e–2 (2.87e–3) −	**6.4459e–2 (2.27e–3)** ≈	6.4604e–2 (2.91e–3) ≈	6.5080e–2 (2.90e–3) ≈	6.4823e–2 (2.19e–3)
+/ − / ≈	0/25/13	2/3/33	3/3/32	2/6/30	

Table 14 The HV values of five algorithms with different gr_1 on three sets of benchmark test functions

Problem	TSRSM3	TSRSM4	TSRSM5	TSRSM6	TSRSM
C1_DTLZ1	8.4142e–1 (1.43e–4) ≈	8.4140e–1 (3.10e–4) ≈	8.4142e–1 (2.19e–4) ≈	8.4136e–1 (2.05e–4) ≈	**8.4144e–1 (1.37e–4)**
C1_DTLZ3	5.4704e–1 (3.59e–2) −	5.5746e–1 (2.23e–3) −	5.5630e–1 (7.88e–3) ≈	5.5792e–1 (2.39e–3) ≈	**5.5858e–1 (2.12e–3)**
C2_DTLZ2	5.1727e–1 (1.09e–3) ≈	5.1743e–1 (1.06e–3) ≈	5.1706e–1 (1.14e–3) ≈	**5.1765e–1 (1.09e–3)** ≈	5.1711e–1 (1.09e–3)
C3_DTLZ4	7.8771e–1 (1.16e–3) ≈	7.8794e–1 (1.39e–3) ≈	7.8802e–1 (1.16e–3) ≈	7.8789e–1 (1.47e–3) ≈	**7.8815e–1 (1.49e–3)**
DC1_DTLZ1	6.3181e–1 (9.30e–4) −	6.3232e–1 (5.88e–4) ≈	6.3221e–1 (5.64e–4) ≈	**6.3233e–1 (5.78e–4)** ≈	6.3224e–1 (6.44e–4)
DC1_DTLZ3	4.8495e–2 (1.25e–1) −	5.2901e–4 (2.90e–3) −	0.0000e+0 (0.00e+0) −	0.0000e+0 (0.00e+0) −	**4.6630e–1 (3.04e–2)**
DC2_DTLZ1	8.2604e–1 (6.86e–2) −	8.4144e–1 (1.83e–4) ≈	8.4147e–1 (1.32e–4) ≈	**8.4153e–1 (1.03e–4)** +	8.4093e–1 (1.58e–3)
DC2_DTLZ3	4.5543e–1 (2.09e–1) ≈	5.0334e–1 (1.67e–1) ≈	**5.5845e–1 (1.90e–3)** ≈	5.4025e–1 (9.96e–2) ≈	4.8035e–1 (1.96e–1)
DC3_DTLZ1	4.4703e–1 (1.86e–1) −	5.3500e–1 (2.11e–3) ≈	5.3521e–1 (3.15e–3) ≈	**5.3583e–1 (9.05e–4)** ≈	5.2057e–1 (7.96e–2)
DC3_DTLZ3	1.2699e–1 (1.72e–1) −	2.7726e–1 (1.56e–1) ≈	**3.5485e–1 (3.47e–2)** +	3.2815e–1 (1.12e–1) +	2.5323e–1 (1.59e–1)
LIRCMOP1	2.2887e–1 (6.41e–3) ≈	2.2926e–1 (6.25e–3) ≈	**2.3099e–1 (3.77e–3)** ≈	2.3059e–1 (3.54e–3) ≈	2.3071e–1 (5.53e–3)
LIRCMOP2	3.5432e–1 (5.87e–3) −	3.5721e–1 (2.38e–3) ≈	3.5715e–1 (2.33e–3) ≈	3.5651e–1 (2.72e–3) −	**3.5740e–1 (2.67e–3)**
LIRCMOP3	1.9660e–1 (7.91e–3) ≈	1.9802e–1 (5.32e–3) ≈	1.9667e–1 (7.32e–3) ≈	1.9649e–1 (6.54e–3) −	**1.9929e–1 (5.84e–3)**
LIRCMOP4	3.0651e–1 (9.38e–3) ≈	3.0639e–1 (7.17e–3) ≈	3.0555e–1 (9.16e–3) ≈	3.0152e–1 (1.81e–2) ≈	**3.0688e–1 (8.24e–3)**
LIRCMOP5	2.8471e–1 (3.57e–2) −	**2.9245e–1 (2.50e–4)** ≈	2.9100e–1 (5.64e–3) ≈	2.9102e–1 (8.00e–3) ≈	2.9068e–1 (9.84e–3)
LIRCMOP6	1.8142e–1 (2.83e–2) −	**1.9799e–1 (1.48e–4)** ≈	1.9796e–1 (1.83e–4) ≈	1.9749e–1 (1.39e–3) ≈	1.9790e–1 (1.68e–4)
LIRCMOP7	2.9356e–1 (3.49e–3) ≈	2.9409e–1 (1.41e–4) ≈	2.9412e–1 (2.04e–4) ≈	2.9380e–1 (1.24e–3) ≈	**2.9415e–1 (1.63e–4)**
LIRCMOP8	2.9412e–1 (5.90e–4) ≈	2.9410e–1 (7.01e–4) ≈	**2.9422e–1 (1.34e–4)** ≈	2.9402e–1 (7.15e–4) ≈	2.9418e–1 (1.10e–4)
LIRCMOP9	5.3023e–1 (1.22e–2) −	5.5230e–1 (9.62e–3) −	5.5004e–1 (1.10e–2) −	5.4761e–1 (1.41e–2) −	**5.5690e–1 (9.27e–3)**
LIRCMOP10	7.0724e–1 (1.67e–3) −	7.0768e–1 (1.41e–4) −	7.0764e–1 (2.35e–4) −	7.0766e–1 (2.33e–4) ≈	**7.0776e–1 (1.73e–4)**
LIRCMOP11	6.8370e–1 (1.66e–2) −	**6.9394e–1 (5.23e–5)** ≈	6.9333e–1 (3.07e–3) +	6.9340e–1 (2.99e–3) ≈	6.9282e–1 (6.16e–3)
LIRCMOP12	6.0884e–1 (1.08e–2) −	6.2009e–1 (8.01e–4) ≈	**6.2028e–1 (7.35e–5)** ≈	6.1990e–1 (1.48e–3) ≈	6.2012e–1 (6.82e–4)
LIRCMOP13	5.5710e–1 (1.14e–3) ≈	5.5700e–1 (1.16e–3) ≈	5.5682e–1 (8.78e–4) ≈	5.5708e–1 (8.12e–4) ≈	**5.5715e–1 (1.02e–3)**
LIRCMOP14	**5.5219e–1 (9.99e–4)** ≈	5.5187e–1 (1.42e–3) ≈	5.5201e–1 (1.20e–3) ≈	5.5204e–1 (1.44e–3) ≈	5.5201e–1 (1.24e–3)
MW1	4.6533e–1 (9.66e–2) −	4.8960e–1 (2.07e–3) ≈	**4.9001e–1 (2.68e–5)** ≈	4.9000e–1 (2.63e–5) ≈	4.9000e–1 (2.25e–5)
MW2	4.7869e–1 (8.85e–2) −	5.6098e–1 (1.18e–2) ≈	**5.6193e–1 (1.23e–2)** ≈	5.6050e–1 (1.36e–2) ≈	5.5704e–1 (1.21e–2)
MW3	5.4427e–1 (4.26e–4) ≈	5.4448e–1 (4.26e–4) ≈	5.4458e–1 (3.59e–4) ≈	**5.4459e–1 (3.13e–4)** ≈	5.4442e–1 (4.68e–4)
MW4	8.3892e–1 (3.72e–3) −	8.4058e–1 (2.40e–3) ≈	8.4102e–1 (3.12e–4) ≈	8.4103e–1 (2.84e–4) ≈	**8.4110e–1 (3.16e–4)**
MW5	2.3911e–1 (1.06e–1) −	**3.2422e–1 (3.19e–4)** ≈	3.2409e–1 (8.50e–4) ≈	3.2073e–1 (1.60e–2) ≈	3.1947e–1 (2.57e–2)
MW6	2.1783e–1 (9.54e–2) −	3.0787e–1 (1.83e–2) ≈	3.0786e–1 (1.22e–2) ≈	**3.1314e–1 (1.01e–2)** ≈	3.0690e–1 (1.51e–2)
MW7	4.1187e–1 (5.43e–4) −	**4.1233e–1 (3.68e–4)** ≈	4.1227e–1 (5.04e–4) ≈	4.1231e–1 (3.18e–4) ≈	4.1224e–1 (3.42e–4)
MW8	4.6342e–1 (8.97e–2) −	**5.4212e–1 (8.88e–3)** ≈	5.4034e–1 (9.31e–3) ≈	5.3672e–1 (1.87e–2) ≈	5.3830e–1 (1.35e–2)
MW9	3.5343e–1 (1.20e–1) −	**3.9871e–1 (1.32e–3)** +	3.9767e–1 (1.91e–3) ≈	3.9821e–1 (1.73e–3) ≈	3.9740e–1 (2.10e–3)
MW10	2.7025e–1 (1.08e–1) −	4.2898e–1 (1.50e–2) ≈	4.1967e–1 (5.91e–2) ≈	4.2808e–1 (1.63e–2) ≈	**4.3043e–1 (1.60e–2)**
MW11	3.9938e–1 (7.42e–2) −	4.3029e–1 (2.95e–2) −	4.2721e–1 (2.87e–2) −	4.2900e–1 (2.22e–2) −	**4.4758e–1 (1.59e–4)**
MW12	5.4922e–1 (1.66e–1) −	6.0421e–1 (3.55e–4) ≈	6.0420e–1 (3.02e–4) ≈	**6.0435e–1 (2.58e–4)** ≈	6.0419e–1 (3.37e–4)
MW13	3.4152e–1 (8.22e–2) −	4.5361e–1 (1.61e–2) ≈	4.5514e–1 (1.96e–2) ≈	4.5621e–1 (1.26e–2) ≈	**4.5873e–1 (1.05e–2)**
MW14	4.7261e–1 (2.19e–3) −	**4.7430e–1 (1.65e–3)** ≈	4.7397e–1 (2.17e–3) ≈	4.7389e–1 (1.80e–3) ≈	4.7407e–1 (1.45e–3)
+/ − / ≈	0/26/12	1/5/32	2/4/32	2/5/31	

Table 15 The IGD+ values of five algorithms with different gr_2 on three sets of benchmark test functions

Problem	TSRSM7	TSRSM8	TSRSM9	TSRSM10	TSRSM
C1_DTLZ1	**1.4685e–2 (7.46e–5)** ≈	1.4716e–2 (8.02e–5) ≈	1.4831e–2 (3.18e–4) −	1.5479e–2 (3.50e–4) −	1.4700e–2 (6.54e–5)
C1_DTLZ3	**2.3405e–2 (7.45e–4)** ≈	2.7265e–2 (7.34e–3) −	2.9210e–1 (1.46e+0) −	2.7007e–2 (1.35e–3) −	2.3835e–2 (1.28e–3)
C2_DTLZ2	1.9151e–2 (5.85e–4) ≈	1.9239e–2 (8.21e–4) ≈	**1.9125e–2 (5.49e–4)** ≈	1.9262e–2 (6.23e–4) ≈	1.9268e–2 (6.03e–4)
C3_DTLZ4	7.0114e–2 (5.95e–2) −	5.8776e–2 (2.54e–3) ≈	5.9820e–2 (2.24e–3) −	6.1780e–2 (2.65e–3) −	**5.8194e–2 (2.25e–3)**
DC1_DTLZ1	8.5642e–3 (2.04e–4) −	**8.4562e–3 (6.52e–5)** ≈	8.5644e–3 (2.19e–4) −	8.8264e–3 (2.03e–4) −	8.4602e–3 (9.26e–5)
DC1_DTLZ3	**1.3425e–2 (3.69e–4)** +	1.4163e–2 (9.84e–4) ≈	1.7965e–2 (1.70e–2) ≈	1.8355e–2 (1.77e–2) −	1.7325e–2 (1.70e–2)
DC2_DTLZ1	**1.4809e–2 (1.39e–4)** ≈	1.5376e–2 (3.33e–3) ≈	2.5271e–2 (3.86e–2) −	2.5627e–2 (3.85e–2) −	1.4953e–2 (7.14e–4)
DC2_DTLZ3	**2.3353e–2 (6.87e–4)** ≈	2.4100e–1 (2.63e–1) −	3.9028e–1 (2.55e–1) −	4.3229e–1 (2.39e–1) −	1.0198e–1 (1.95e–1)
DC3_DTLZ1	**5.1971e–3 (1.36e–4)** ≈	6.3015e–3 (3.11e–3) ≈	5.5413e–2 (1.20e–1) −	6.9308e–2 (1.20e–1) −	1.0792e–2 (3.00e–2)
DC3_DTLZ3	**8.3950e–3 (4.73e–4)** +	6.7723e–1 (5.40e–1) −	8.5488e–1 (4.11e–1) −	8.9926e–1 (5.61e–1) −	1.7186e–1 (2.60e–1)
LIRCMOP1	1.5398e–1 (6.10e–2) −	1.1288e–2 (4.23e–3) ≈	1.3108e–2 (7.00e–3) ≈	**1.0205e–2 (3.44e–3)** +	1.4963e–2 (8.07e–3)
LIRCMOP2	7.5101e–2 (3.39e–2) −	8.4470e–3 (3.11e–3) ≈	8.0202e–3 (3.92e–3) ≈	**7.5491e–3 (1.84e–3)** ≈	9.0102e–3 (4.33e–3)
LIRCMOP3	1.5870e–1 (6.05e–2) −	**1.3110e–2 (8.21e–3)** ≈	1.4286e–2 (1.72e–2) ≈	1.3535e–2 (9.60e–3) ≈	1.4317e–2 (8.99e–3)
LIRCMOP4	1.1236e–1 (4.34e–2) −	1.0961e–2 (8.47e–3) ≈	1.0557e–2 (8.22e–3) ≈	**8.4999e–3 (6.52e–3)** +	1.5124e–2 (1.12e–2)
LIRCMOP5	1.1167e–2 (3.36e–3) −	3.3766e–3 (2.74e–4) +	3.2720e–3 (1.12e–4) +	**3.2667e–3 (1.01e–4)** +	5.6297e–3 (1.02e–2)
LIRCMOP6	1.1114e–2 (5.24e–3) −	2.8137e–3 (8.17e–5) +	**2.7687e–3 (3.81e–5)** +	2.7816e–3 (5.20e–5) +	3.3155e–3 (3.19e–4)
LIRCMOP7	8.9531e–3 (4.03e–3) −	6.3551e–3 (4.06e–4) ≈	6.3488e–3 (3.70e–4) ≈	**6.2294e–3 (2.40e–4)** +	6.5277e–3 (3.74e–4)
LIRCMOP8	7.3868e–3 (2.92e–3) −	5.7139e–3 (3.10e–4) +	5.6742e–3 (2.55e–4) +	**5.6095e–3 (2.68e–4)** +	6.4708e–3 (2.57e–4)
LIRCMOP9	2.2328e–2 (1.90e–2) ≈	2.5454e–2 (2.51e–2) ≈	2.1247e–2 (2.90e–2) ≈	**1.4189e–2 (2.29e–2)** +	2.2200e–2 (2.23e–2)
LIRCMOP10	5.4982e–3 (5.26e–4) −	3.2317e–e–3 (1.75e–4) +	**3.0789e–3 (1.29e–4)** +	3.1104e–3 (1.24e–4) +	4.1256e–3 (2.73e–4)
LIRCMOP11	7.1358e–4 (1.43e–4) ≈	2.6725e–3 (9.20e–3) −	**7.1073e–4 (7.68e–5)** +	1.3888e–3 (2.48e–3) ≈	2.3187e–3 (9.08e–3)
LIRCMOP12	1.0801e–3 (1.50e–3) −	8.4276e–4 (2.07e–3) ≈	6.4297e–4 (1.12e–3) ≈	**4.3186e–4 (4.92e–5)** ≈	7.9006e–4 (1.33e–3)
LIRCMOP13	4.2497e–2 (1.39e–3) ≈	4.2342e–2 (1.10e–3) ≈	4.2460e–2 (1.50e–3) ≈	4.2386e–2 (1.38e–3) ≈	**4.2186e–2 (1.16e–3)**
LIRCMOP14	4.8549e–2 (1.75e–3) ≈	**4.7869e–2 (1.23e–3)** ≈	4.7878e–2 (1.39e–3) ≈	4.7898e–2 (1.19e–3) ≈	4.8073e–2 (1.23e–3)
MW1	1.2524e–3 (1.32e–5) ≈	1.2857e–3 (3.92e–5) −	1.3421e–3 (3.72e–5) −	1.3595e–3 (4.04e–5) −	**1.2487e–3 (1.08e–5)**
MW2	**1.4509e–2 (5.87e–3)** ≈	1.5210e–2 (7.16e–3) ≈	1.6176e–2 (6.54e–3) ≈	1.7052e–2 (9.15e–3) ≈	1.7687e–2 (7.60e–3)
MW3	**2.8687e–3 (2.65e–4)** ≈	2.9681e–3 (2.67e–4) ≈	2.9282e–3 (1.64e–4) ≈	2.9944e–3 (2.99e–4) ≈	2.9021e–3 (2.79e–4)
MW4	**2.9703e–2 (2.29e–4)** ≈	2.9778e–2 (2.97e–4) ≈	3.0696e–2 (1.50e–3) −	3.1057e–2 (9.15e–4) −	2.9735e–2 (2.24e–4)
MW5	1.5673e–3 (1.56e–3) ≈	1.8527e–3 (4.18e–3) ≈	1.4331e–3 (1.27e–3) ≈	**1.1408e–3 (8.61e–4)** ≈	6.6809e–3 (3.13e–2)
MW6	1.5434e–2 (1.08e–2) ≈	1.8230e–2 (9.11e–3) ≈	**1.3843e–2 (1.42e–2)** ≈	1.4774e–2 (1.07e–2) ≈	1.6864e–2 (1.11e–2)
MW7	**2.0251e–3 (1.53e–4)** ≈	2.2694e–3 (2.75e–4) −	2.2617e–3 (2.03e–4) −	2.3227e–3 (2.10e–4) −	2.0949e–3 (1.60e–4)
MW8	2.9867e–2 (8.73e–3) ≈	2.8229e–2 (6.86e–3) ≈	2.7559e–2 (5.73e–3) ≈	2.8265e–2 (5.31e–3) ≈	**2.7142e–2 (7.45e–3)**
MW9	**3.2370e–3 (1.99e–4)** +	3.5324e–3 (4.19e–4) ≈	4.7667e–3 (2.41e–3) −	4.2447e–3 (9.35e–4) −	3.3726e–3 (2.62e–4)
MW10	2.4790e–2 (1.89e–2) ≈	2.6094e–2 (2.06e–2) ≈	**1.6866e–2 (1.07e–2)** ≈	2.7775e–2 (2.30e–2) ≈	2.2101e–2 (1.71e–2)
MW11	**2.8528e–3 (1.90e–4)** +	3.2457e–3 (2.13e–4) −	3.4033e–3 (2.03e–4) −	3.3986e–3 (1.62e–4) −	2.9800e–3 (1.60e–4)
MW12	3.4493e–3 (7.09e–4) ≈	2.4508e–2 (1.16e–1) ≈	2.7882e–2 (1.34e–1) ≈	3.6102e–3 (1.79e–4) −	**3.3639e–3 (1.75e–4)**
MW13	**2.2437e–2 (1.29e–2)** ≈	2.6561e–2 (1.55e–2) ≈	2.5641e–2 (1.08e–2) ≈	3.1819e–2 (2.14e–2) ≈	2.3501e–2 (1.21e–2)
MW14	**6.4001e–2 (2.65e–3)** ≈	6.9022e–2 (4.64e–3) −	7.0561e–2 (3.54e–3) −	7.1480e–2 (2.85e–3) −	6.4823e–2 (2.19e–3)
+/ − / ≈	4/12/22	4/8/26	5/14/19	8/16/14	

Table 16 The HV values of five algorithms with different gr_2 on three sets of benchmark test functions

Problem	TSRSM7	TSRSM8	TSRSM9	TSRSM10	TSRSM
C1_DTLZ1	8.4073e–1 (7.45e–4) −	8.4134e–1 (2.11e–4) ≈	8.4088e–1 (1.10e–3) −	8.3799e–1 (1.21e–3) −	**8.4144e–1 (1.37e–4)**
C1_DTLZ3	**5.5914e–1 (1.08e–3)** ≈	5.5247e–1 (1.32e–2) −	5.3629e–1 (1.01e–1) −	5.5284e–1 (2.02e–3) −	5.5858e–1 (2.12e–3)
C2_DTLZ2	5.1712e–1 (1.27e–3) ≈	5.1710e–1 (1.48e–3) ≈	**5.1756e–1 (1.26e–3)** ≈	5.1662e–1 (1.64e–3) ≈	5.1711e–1 (1.09e–3)
C3_DTLZ4	7.7926e–1 (4.56e–2) ≈	7.8785e–1 (1.53e–3) ≈	7.8712e–1 (1.45e–3) −	7.8589e–1 (1.66e–3) −	**7.8815e–1 (1.49e–3)**
DC1_DTLZ1	6.3174e–1 (6.81e–4) −	6.3200e–1 (5.62e–4) −	6.3177e–1 (1.16e–3) ≈	6.3054e–1 (8.99e–4) −	**6.3224e–1 (6.44e–4)**
DC1_DTLZ3	**4.7360e–1 (7.85e–4)** +	4.7135e–1 (2.58e–3) ≈	4.6442e–1 (3.02e–2) −	4.6315e–1 (3.18e–2) −	4.6630e–1 (3.04e–2)
DC2_DTLZ1	**8.4124e–1 (2.52e–4)** +	8.3998e–1 (7.37e–3) ≈	8.1567e–1 (9.42e–2) −	8.1471e–1 (9.48e–2) −	8.4093e–1 (1.58e–3)
DC2_DTLZ3	**5.5965e–1 (9.86e–4)** ≈	3.3270e–1 (2.61e–1) −	1.8603e–1 (2.52e–1) −	1.4735e–1 (2.38e–1) −	4.8035e–1 (1.96e–1)
DC3_DTLZ1	**5.3549e–1 (1.06e–3)** ≈	5.3031e–1 (1.49e–2) ≈	4.4287e–1 (1.90e–1) −	4.0966e–1 (2.13e–1) −	5.2057e–1 (7.96e–2)
DC3_DTLZ3	**3.6761e–1 (1.02e–3)** +	6.9662e–2 (1.43e–1) −	1.2242e–2 (6.71e–2) −	2.3800e–2 (9.06e–2) −	2.5323e–1 (1.59e–1)
LIRCMOP1	1.5254e–1 (2.30e–2) −	2.3305e–1 (2.81e–3) ≈	2.3209e–1 (4.40e–3) ≈	**2.3386e–1 (2.10e–3)** +	2.3071e–1 (5.53e–3)
LIRCMOP2	2.9180e–1 (2.93e–2) −	3.5779e–1 (1.85e–3) ≈	3.5803e–1 (2.43e–3) ≈	**3.5829e–1 (1.22e–3)** ≈	3.5740e–1 (2.67e–3)
LIRCMOP3	1.3465e–1 (2.30e–2) −	**2.0030e–1 (5.34e–3)** ≈	1.9985e–1 (9.57e–3) ≈	2.0025e–1 (6.79e–3) ≈	1.9929e–1 (5.84e–3)
LIRCMOP4	2.3608e–1 (2.74e–2) −	3.0986e–1 (5.79e–3) ≈	3.1001e–1 (6.96e–3) ≈	**3.1172e–1 (4.44e–3)** +	3.0688e–1 (8.24e–3)
LIRCMOP5	2.8653e–1 (3.40e–3) −	2.9268e–1 (1.46e–4) +	2.9274e–1 (6.10e–5) +	**2.9274e–1 (5.46e–5)** +	2.9068e–1 (9.84e–3)
LIRCMOP6	1.9331e–1 (2.82e–3) −	1.9817e–1 (4.48e–5) +	**1.9819e–1 (2.66e–5)** +	1.9818e–1 (3.23e–5) +	1.9790e–1 (1.68e–4)
LIRCMOP7	2.9202e–1 (3.24e–3) −	2.9424e–1 (1.78e–4) ≈	2.9424e–1 (1.35e–4) +	**2.9429e–1 (9.11e–5)** +	2.9415e–1 (1.63e–4)
LIRCMOP8	2.9336e–1 (2.37e–3) −	2.9453e–1 (1.25e–4) +	2.9456e–1 (1.22e–4) +	**2.9458e–1 (1.21e–4)** +	2.9418e–1 (1.10e–4)
LIRCMOP9	5.5681e–1 (7.63e–3) ≈	5.5599e–1 (1.13e–2) ≈	5.5688e–1 (1.46e–2) ≈	**5.6055e–1 (1.11e–2)** +	5.5690e–1 (9.27e–3)
LIRCMOP10	7.0631e–1 (4.80e–4) −	7.0821e–1 (1.33e–4) +	**7.0828e–1 (8.21e–5)** +	7.0825e–1 (1.12e–4) +	7.0776e–1 (1.73e–4)
LIRCMOP11	6.9389e–1 (1.07e–4) +	6.9256e–1 (6.27e–3) ≈	**6.9395e–1 (4.25e–5)** ≈	6.9347e–1 (1.66e–3) ≈	6.9282e–1 (6.16e–3)
LIRCMOP12	6.2002e–1 (6.17e–4) −	6.2011e–1 (1.04e–3) −	6.2019e–1 (5.81e–4) +	**6.2029e–1 (3.04e–5)** ≈	6.2012e–1 (6.82e–4)
LIRCMOP13	5.5690e–1 (1.40e–3) ≈	5.5711e–1 (9.00e–4) ≈	5.5682e–1 (1.60e–3) ≈	5.5698e–1 (1.15e–3) ≈	**5.5715e–1 (1.02e–3)**
LIRCMOP14	5.5156e–1 (1.49e–3) ≈	5.5225e–1 (1.13e–3) ≈	5.5238e–1 (1.25e–3) ≈	**5.5240e–1 (1.13e–3)** ≈	5.5201e–1 (1.24e–3)
MW1	**4.9001e–1 (2.59e–5)** ≈	4.8990e–1 (5.88e–5) −	4.8985e–1 (6.13e–5) −	4.8980e–1 (7.40e–5) −	4.9000e–1 (2.25e–5)
MW2	**5.6174e–1 (1.01e–2)** ≈	5.6084e–1 (1.18e–2) ≈	5.5903e–1 (1.05e–2) ≈	5.5802e–1 (1.44e–2) ≈	5.5704e–1 (1.21e–2)
MW3	5.4441e–1 (5.02e–4) ≈	5.4434e–1 (4.54e–4) ≈	5.4441e–1 (2.74e–4) ≈	5.4426e–1 (4.98e–4) ≈	**5.4442e–1 (4.68e–4)**
MW4	**8.4112e–1 (2.18e–4)** ≈	8.4102e–1 (3.78e–4) ≈	8.3981e–1 (1.91e–3) −	8.3949e–1 (1.02e–3) −	8.4110e–1 (3.16e–4)
MW5	3.2388e–1 (8.69e–4) ≈	3.2342e–1 (3.38e–3) ≈	3.2387e–1 (7.45e–4) ≈	**3.2406e–1 (3.94e–4)** ≈	3.1947e–1 (2.57e–2)
MW6	3.0884e–1 (1.44e–2) ≈	3.0500e–1 (1.24e–2) ≈	**3.1155e–1 (1.64e–2)** ≈	3.0972e–1 (1.46e–2) ≈	3.0690e–1 (1.51e–2)
MW7	**4.1236e–1 (3.22e–4)** ≈	4.1214e–1 (5.26e–4) ≈	4.1216e–1 (3.41e–4) ≈	4.1212e–1 (4.10e–4) ≈	4.1224e–1 (3.42e–4)
MW8	5.3356e–1 (1.58e–2) ≈	5.3634e–1 (1.22e–2) ≈	5.3766e–1 (1.03e–2) ≈	5.3594e–1 (9.53e–3) ≈	**5.3830e–1 (1.35e–2)**
MW9	**3.9863e–1 (1.87e–3)** +	3.9714e–1 (2.50e–3) ≈	3.9362e–1 (5.28e–3) −	3.9420e–1 (3.10e–3) −	3.9740e–1 (2.10e–3)
MW10	4.2781e–1 (1.70e–2) ≈	4.2697e–1 (1.85e–2) ≈	**4.3523e–1 (1.17e–2)** ≈	4.2558e–1 (1.98e–2) ≈	4.3043e–1 (1.60e–2)
MW11	4.4671e–1 (1.91e–3) −	4.4750e–1 (1.84e–4) ≈	4.4755e–1 (1.54e–4) ≈	4.4753e–1 (1.39e–4) ≈	**4.4758e–1 (1.59e–4)**
MW12	6.0412e–1 (1.01e–3) ≈	5.8484e–1 (1.05e–1) −	5.8389e–1 (1.10e–1) −	6.0381e–1 (2.77e–4) −	**6.0419e–1 (3.37e–4)**
MW13	**4.5962e–1 (1.09e–2)** ≈	4.5637e–1 (1.23e–2) ≈	4.5665e–1 (8.80e–3) ≈	4.5253e–1 (1.59e–2) ≈	4.5873e–1 (1.05e–2)
MW14	**4.7446e–1 (1.54e–3)** ≈	4.7151e–1 (3.15e–3) −	4.7011e–1 (2.64e–3) −	4.6941e–1 (1.92e–3) −	4.7407e–1 (1.45e–3)
+/ − / ≈	5/13/20	4/8/26	6/13/19	8/14/16	

Appendix B

See Fig. 6.

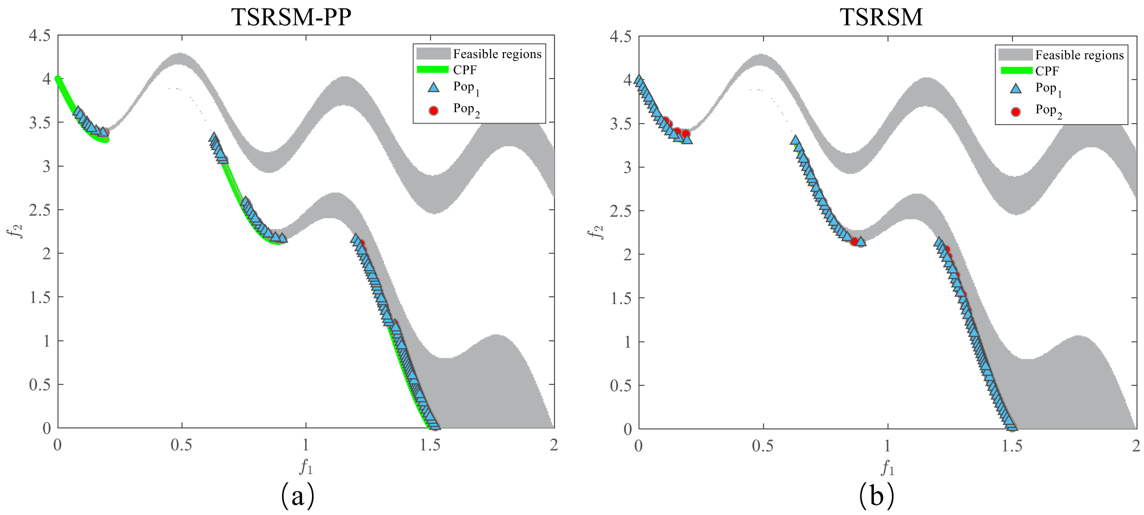


Fig. 6 The distribution of populations on MW13. **a** TSRSM-PP. **b** TSRSM

References

1. Peng C, Qiu S (2022) A decomposition-based constrained multi-objective evolutionary algorithm with a local infeasibility utilization mechanism for UAV path planning. Appl Soft Comput 118:108495. https://doi.org/10.1016/j.asoc.2022.108495
2. Rani D, Ebrahimnejad A, Gupta G (2022) Generalized techniques for solving intuitionistic fuzzy multi-objective non-linear optimization problems. Expert Syst Appl 202:117264. https://doi.org/10.1016/j.eswa.2022.117264
3. Ming F, Gong W, Yang Y, Liao Z (2023) Constrained multimodal multi-objective optimization: Test problem construction and algorithm design. Swarm Evol Comput 76:101209. https://doi.org/10.1016/j.swevo.2022.101209
4. Bind AK, Rani D, Goyal KK, Ebrahimnejad A (2023) A solution approach for sustainable multi-objective multi-item 4d solid transportation problem involving triangular intuitionistic fuzzy parameters. J Clean Prod. https://doi.org/10.1016/j.jclepro.2023.137661
5. Ming M, Wang R, Ishibuchi H, Zhang T (2022) A novel dual-stage dual-population evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 26(5):1129–1143
6. Sharma K, Singh VP, Ebrahimnejad A, Chakraborty D (2023) Solving a multi-objective chance constrained hierarchical optimization problem under intuitionistic fuzzy environment with its application. Expert Syst Appl 217:119595. https://doi.org/10.1016/j.eswa.2023.119595
7. Singh V, Sharma K, Singh B, Ebrahimnejad A, Chakraborty D (2023) Fermatean fuzzy vehicle routing problem with profit: solution algorithms, comparisons and developments. Sādhanā 48(3):166. https://doi.org/10.1007/s12046-023-02238-5
8. Ma Z, Wang Y (2023) Shift-based penalty for evolutionary constrained multiobjective optimization and its application. IEEE Trans Cybern 53(1):18–30
9. Fan Z, Ruan J, Li W, You Y, Cai X, Xu Z, Yang Z, Sun F, Wang Z, Yuan Y et al (2019) A learning guided parameter setting for constrained multi-objective optimization. In: 2019 1st international conference on industrial artificial intelligence (IAI). IEEE, pp 1–6
10. Yang Y, Liu J, Tan S (2020) A constrained multi-objective evolutionary algorithm based on decomposition and dynamic constraint-handling mechanism. Appl Soft Comput 89:106104. https://doi.org/10.1016/j.asoc.2020.106104
11. Gu Q, Wang Q, Xiong NN, Jiang S, Chen L (2022) Surrogate-assisted evolutionary algorithm for expensive constrained multi-objective discrete optimization problems. Complex Intell Syst 8:2699–2718
12. Zhou Y, Zhu M, Wang J, Zhang Z, Xiang Y, Zhang J (2018) Tri-goal evolution framework for constrained many-objective optimization. IEEE Trans Syst Man Cybern Syst 50(8):3086–3099
13. Long Q (2014) A constraint handling technique for constrained multi-objective genetic algorithm. Swarm Evol Comput 15:66–79
14. Uribe L, Lara A, Schütze O (2020) On the efficient computation and use of multi-objective descent directions within constrained MOEAs. Swarm Evol Comput 52:100617. https://doi.org/10.1016/j.swevo.2019.100617
15. Hernández VAS, Schütze O, Wang H, Deutz A, Emmerich M (2018) The set-based hypervolume newton method for bi-objective optimization. IEEE Trans Cybern 50(5):2186–2196
16. Yu K, Liang J, Qu B, Luo Y, Yue C (2022) Dynamic selection preference-assisted constrained multiobjective differential evolution. IEEE Trans Syst Man Cybern Syst 52(5):2954–2965
17. Long W, Dong H, Wang P, Huang Y, Li J, Yang X, Fu C (2023) A constrained multi-objective optimization algorithm using an efficient global diversity strategy. Complex Intell Syst 9(2):1455–1478
18. Rathore AK, Holtz J, Boller T (2010) Synchronous optimal pulsewidth modulation for low-switching-frequency control of medium-voltage multilevel inverters. IEEE Trans Ind Electron 57(7):2374–2381
19. Liang J, Ban X, Yu K, Qu B, Qiao K, Yue C, Chen K, Tan KC (2023) A survey on evolutionary constrained multiobjective optimization. IEEE Trans Evol Comput 27(2):201–221
20. Song X, Wu N, Song S, Stojanovic V (2023) Switching-like event-triggered state estimation for reaction-diffusion neural networks against DoS attacks. Neural Process Lett 55:8997–9018. https://doi.org/10.1007/s11063-023-11189-1
21. Kumar A, Wu G, Ali MZ, Luo Q, Mallipeddi R, Suganthan PN, Das S (2021) A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results. Swarm Evol Comput 67:100961. https://doi.org/10.1016/j.swevo.2021.100961
22. Dhiman G, Kumar V (2018) Multi-objective spotted hyena optimizer: a multi-objective optimization algorithm for engineering problems. Knowl Based Syst 150:175–197

23. Guillén-Gosálbez G (2011) A novel milp-based objective reduction method for multi-objective optimization: application to environmental problems. Comput Chem Eng 35(8):1469–1477
24. Sun R, Zou J, Liu Y, Yang S, Zheng J (2023) A multistage algorithm for solving multiobjective optimization problems with multiconstraints. IEEE Trans Evol Comput 27(5):1207–1219
25. Dong J, Gong W, Ming F (2023) A tri-stage competitive swarm optimizer for constrained multi-objective optimization. Appl Intell 53(7):7892–7916
26. Tian Y, Zhang T, Xiao J, Zhang X, Jin Y (2021) A coevolutionary framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput 25(1):102–116
27. Ming M, Trivedi A, Wang R, Srinivasan D, Zhang T (2021) A dual-population-based evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 25(4):739–753
28. Fan Z, Li W, Cai X, Li H, Wei C, Zhang Q, Deb K, Goodman E (2019) Push and pull search for solving constrained multi-objective optimization problems. Swarm Evol Comput 44:665–679
29. Tian Y, Zhang Y, Su Y, Zhang X, Tan KC, Jin Y (2021) Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization. IEEE Trans Cybern 52(9):9559–9572
30. Dong J, Gong W, Ming F, Wang L (2022) A two-stage evolutionary algorithm based on three indicators for constrained multi-objective optimization. Expert Syst Appl 195:116499. https://doi.org/10.1016/j.eswa.2022.116499
31. Wang Y, Liu Y, Zou J, Zheng J, Yang S (2022) A novel two-phase evolutionary algorithm for solving constrained multi-objective optimization problems. Swarm Evol Comput 75:101166. https://doi.org/10.1016/j.swevo.2022.101166
32. Liang J, Qiao K, Yu K, Qu B, Yue C, Guo W, Wang L (2023) Utilizing the relationship between unconstrained and constrained pareto fronts for constrained multiobjective optimization. IEEE Trans Cybern 53(6):3873–3886
33. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C (2022) An evolutionary multitasking optimization framework for constrained multiobjective optimization problems. IEEE Trans Evol Comput 26(2):263–277
34. Chen K, Xue B, Zhang M, Zhou F (2020) An evolutionary multitasking-based feature selection method for high-dimensional classification. IEEE Trans Cybern 52(7):7172–7186
35. Qiao K, Yu K, Qu B, Liang J, Song H, Yue C, Lin H, Tan KC (2023) Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization. IEEE Trans Evol Comput 27(3):642–656
36. Ming F, Gong W, Wang L, Gao L (2024) Constrained multi-objective optimization via multitasking and knowledge transfer. IEEE Trans Evol Comput 28(1):77–89
37. Ming F, Gong W, Wang L, Lu C (2022) A tri-population based co-evolutionary framework for constrained multi-objective optimization problems. Swarm Evol Comput 70:101055. https://doi.org/10.1016/j.swevo.2022.101055
38. Ma Z, Wang Y (2019) Evolutionary constrained multiobjective optimization: test suite construction and performance comparisons. IEEE Trans Evol Comput 23(6):972–986
39. Fan Z, Li W, Cai X, Huang H, Fang Y, You Y, Mo J, Wei C, Goodman E (2019) An improved epsilon constraint-handling method in MOEA/D for cmops with large infeasible regions. Soft Comput 23:12491–12510
40. Zitzler E, Laumanns M, Thiele L (2001) SPEA2: improving the strength pareto evolutionary algorithm. TIK-report 103. https://doi.org/10.3929/ethz-a-004284029
41. Deb K, Agrawal RB (1995) Simulated binary crossover for continuous search space. Complex Syst 9(2):115–148
42. Deb K, Goyal M (1996) A combined genetic adaptive search (geneAS) for engineering design. Comput Sci Inform 26:30–45
43. Tian Y, Cheng R, Zhang X, Jin Y (2017) Platemo: a MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Comput Intell Mag 12(4):73–87
44. Jain H, Deb K (2014) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 18(4):602–622
45. Li K, Chen R, Fu G, Yao X (2019) Two-archive evolutionary algorithm for constrained multiobjective optimization. IEEE Trans Evol Comput 23(2):303–315
46. Jain H, Deb K (2013) An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach. IEEE Trans Evol Comput 18(4):602–622
47. Ishibuchi H, Masuda H, Tanigaki Y, Nojima Y (2015) Modified distance calculation in generational distance and inverted generational distance. In: Gaspar-Cunha A, Henggeler Antunes C, Coello CC (eds) Evolutionary multi-criterion optimization. Springer, Cham, pp 110–125
48. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans Evol Comput 3(4):257–271
49. Liu Z-Z, Wang Y (2019) Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. IEEE Trans Evol Comput 23(5):870–884
50. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. J Heurist 15(6):617–644
51. Zhuang Z, Tao H, Chen Y, Stojanovic V, Paszke W (2023) An optimal iterative learning control approach for linear systems with nonuniform trial lengths under input constraints. IEEE Trans Syst Man Cybern Syst 53(6):3461–3473
52. Peng Z, Song X, Song S, Stojanovic V (2023) Hysteresis quantified control for switched reaction-diffusion systems and its application. Complex Intell Syst 9:7451–7460
53. Di Caprio D, Ebrahimnejad A, Alrezaamiri H, Santos-Arteaga FJ (2022) A novel ant colony algorithm for solving shortest path problems with fuzzy arc weights. Alex Eng J 61(5):3403–3415
54. Zhang Z, Song X, Sun X, Stojanovic V (2023) Hybrid-driven-based fuzzy secure filtering for nonlinear parabolic partial differential equation systems with cyber attacks. Int J Adapt Control Signal Process 37(2):380–398
55. Sharma K, Singh VP, Poojara B, Ebrahimnejad A, Chakraborty D (2023) An optimization method to solve a fully intuitionistic fuzzy non-linear separable programming problem. RAIRO Oper Res 57(6):3117–3139