


## Article

# A Zoning Search-Based Multimodal Multi-Objective Brain Storm Optimization Algorithm for Multimodal Multi-Objective Optimization

Jiajia Fan <sup>1</sup>, Wentao Huang <sup>2</sup>, Qingchao Jiang <sup>3</sup>  and Qinqin Fan <sup>1,\*</sup>

<sup>1</sup> Logistics Research Center, Shanghai Maritime University, Shanghai 201306, China; 202030510156@stu.shmtu.edu.cn

<sup>2</sup> Key Laboratory of Control of Power Transmission and Conversion, Ministry of Education, Shanghai Jiao Tong University, Shanghai 200240, China; hwt8989@sjtu.edu.cn

<sup>3</sup> Key Laboratory of Smart Manufacturing in Energy Chemical, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China; qchjiang@ecust.edu.cn

\* Correspondence: qqfan@shmtu.edu.cn

**Abstract:** For multimodal multi-objective optimization problems (MMOPs), there are multiple equivalent Pareto optimal solutions in the decision space that are corresponding to the same objective value. Therefore, the main tasks of multimodal multi-objective optimization (MMO) are to find a high-quality PF approximation in the objective space and maintain the population diversity in the decision space. To achieve the above objectives, this article proposes a zoning search-based multimodal multi-objective brain storm optimization algorithm (ZS-MMBSO). At first, the search space segmentation method is employed to divide the search space into some sub-regions. Moreover, a novel individual generation strategy is incorporated into the multimodal multi-objective brain storm optimization algorithm, which can improve the search performance of the search engineering. The proposed algorithm is compared with five famous multimodal multi-objective evolutionary algorithms (MMOEA) on IEEE CEC2019 MMOPs benchmark test suite. Experimental results indicate that the overall performance of the ZS-MMBSO is the best among all competitors.



**Citation:** Fan, J.; Huang, W.; Jiang, Q.; Fan, Q. A Zoning Search-Based Multimodal Multi-Objective Brain Storm Optimization Algorithm for Multimodal Multi-Objective Optimization. *Algorithms* **2023**, *16*, 350. <https://doi.org/10.3390/a16070350>

Academic Editor: Maciej Drozdowski

Received: 3 June 2023

Revised: 16 July 2023

Accepted: 17 July 2023

Published: 21 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** brain storm optimization algorithm; multimodal multi-objective optimization; evolutionary computation; zoning search

## 1. Introduction

Multimodal multi-objective optimization problems (MMOPs) have become a research hot spot in recent years. As we know, there are at least two conflicting objectives in multi-objective optimization problems (MOPs) [1–3]. However, for actual problems, some of them have multiple equivalent Pareto optimal solutions in the decision space that are corresponding to the same objective value. These special MOPs are termed as MMOPs [4]. Clearly, how to maintain the population diversity in the decision space and find the high-quality Pareto front (PF) approximation in the objective space are two challenging tasks.

In order to effectively solve MMOPs, the niching methods [5–8], the diversity maintenance technique [9], and multiple population strategies [10,11] have been incorporated into multi-objective evolutionary algorithms (MOEAs), such as the decision space-based niching NSGAI (DN-NSGAI) [4] and the multi-objective particle swarm optimization (PSO) based on the self-organizing mechanism (SMP SO-MM) [12]. Besides the above “soft isolation” methods, the “hard isolation” methods (i.e., Zoning search (ZS)) [13] is employed to reduce the solution complexity of MMOPs. Subsequently, the ZS with an adaptive resource-allocating method is proposed by Fan et al. [14], which can further improve the solution efficiency of the ZS. From the above studies, it is clear that the previous studies focused on how to preserve the equivalent Pareto optimal solutions. Additionally, search

engineering is also important for solving MMOPs as it can greatly affect the population diversity throughout the evolution. Like other meta-heuristics algorithms, brain storm optimization (BSO) [15,16] has good global exploration ability and is suitable for solving different types of MOPs. Moreover, existing studies [17–21] show that the BSO is a competitive method to solve multimodal optimization problems. Therefore, the BSO may be an effective method to solve MMOPs.

In the present study, a zoning search-based multimodal multi-objective brain storm optimization (ZS-MMBSO) is proposed. In the ZS-MMBSO, the ZS [13] is used to divide the decision space, which can reduce the search complexity and improve the population diversity. In addition, a novel individual generation strategy is proposed which can enhance the exploration capability in the early stage and the exploitation ability in the later stage. Note that a “soft isolation” method (i.e., K-means) is used in the BSO. Therefore, the proposed algorithm uses both “soft isolation” and “hard isolation” to locate and preserve the equivalent solutions. In this paper, five famous multimodal multi-objective evolutionary algorithms (MMOEs) are chosen to compare with the ZS-MMBSO on 22 MMOPs introduced in IEEE CEC2019 [22]. The experimental results indicate that compared with other well-known MMOEs, the ZS-MMBSO is competitive in solving MMOPs.

The remaining parts of this article are as follows: The basic concepts of the MOPs, the performance metric, and the BSO are introduced in Section 2. In Section 3, the relevant work in the field of MMO is reviewed. Section 4 shows the details of the ZS-MMBSO. And in Section 5, the experimental results and analyses are reported. Section 6 summarizes the conclusions.

## 2. Preliminaries

### 2.1. Multi-objective Optimization Problems

The definition of the MOPs is as follows [1–3]:

$$\begin{aligned} \min_{\mathbf{x} \in \Omega} \mathbf{F}(\mathbf{x}) &= [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^T, \\ \text{subject to } \mathbf{x}_i &\in (x_i^{low}, x_i^{high}), i = 1, 2, \dots, n, \end{aligned} \quad (1)$$

where  $\mathbf{x}$  is an  $n$ -dimensional decision vector in the decision space  $\Omega$ ;  $\mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^T$  is an objective vector containing  $m$  objectives in the objective space.  $x_i^{low}$  and  $x_i^{high}$  are the boundaries of the  $i$ -th decision vector, respectively.

The following are the basic concepts of MOPs:

**Definition 1.** (Pareto Dominance): For the vector  $\mathbf{u} = [u_1, u_2, \dots, u_m]^T$  and  $\mathbf{v} = [v_1, v_2, \dots, v_m]^T$ , if  $\forall j \in \{1, 2, \dots, m\}$ ,  $u_j \leq v_j$ , and  $\exists s \in \{1, 2, \dots, m\}$ ,  $u_s < v_s$ , then the vector  $\mathbf{u}$  dominates another vector  $\mathbf{v}$ , expressed as  $\mathbf{u} \prec \mathbf{v}$ .

**Definition 2.** (Pareto Optimal Set): For a solution  $\mathbf{x}^* \in \Omega$ , if there is no other solution  $\mathbf{x} \in \Omega$  which satisfied  $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x}^*)$ ,  $\mathbf{x}^*$  is called the Pareto optimal solution. All Pareto optimal solutions constitute PS, denoted by  $\mathbf{X}^*$ .

**Definition 3.** (Pareto Front): All objective vectors corresponding to PS form Pareto front (PF), denoted as  $PF = \{\mathbf{F}(\mathbf{x}^*) | \mathbf{x}^* \in \mathbf{X}^*\}$

### 2.2. Performance Metric

To demonstrate the effectiveness of MMOEs, various performance metrics have been proposed, such as the Pareto set proximity (PSP) [23] and the hypervolume (HV) [24].

(1) The PSP is calculated as follows:

$$\begin{aligned}
 PSP &= \frac{CR}{IGD_x}, \\
 IGD_x(A, PS^*) &= \frac{\sum_{v \in PS^*} d(v, A)}{|PS^*|}, \\
 CR &= \left( \prod_{j=1}^D \delta_j \right)^{\frac{1}{D}}, \\
 \delta_j &= \begin{cases} 1, & V_j^{\max} = V_j^{\min} \\ 0, & v_j^{\min} \geq V_j^{\max} \parallel v_j^{\max} \leq V_j^{\min}, \\ \left( \frac{\min(v_j^{\max}, V_j^{\max}) - \max(v_j^{\min}, V_j^{\min})}{V_j^{\max} - V_j^{\min}} \right), & otherwise \end{cases}
 \end{aligned} \tag{2}$$

where  $CR$  is the cover rate and  $IGD_x$  represent the inverted generational distance in the decision space;  $PS^*$  and  $A$  represent the true PS and the obtained PS, respectively; The Euclidean distance between  $v$  and solutions in  $A$  is expressed as  $d(v, A)$ ; And the number of solutions in  $PS^*$  is denoted as  $|PS^*|$ ; For the  $j$ -th variable,  $V_j^{\max}$  and  $v_j^{\max}$  denote the maximum in  $PS^*$  and  $A$ ,  $V_j^{\min}$ , and  $v_j^{\min}$  are the minimum in  $PS^*$  and  $A$ . A larger  $PSP$  means the algorithm performs better in the decision space.

(2) The calculation method for  $HV$  is as follows:

$$HV(PS) = VOL(\cup_{x \in PS^*} [f_1(x), r_1^*] \times \dots \times [f_m(x), r_m^*]), \tag{3}$$

where  $VOL()$  is the Lebesgue measure;  $r^* = (r_1^*, r_2^*, \dots, r_m^*)$  represents the reference point in the objective space. The  $HV$  is used to evaluate the performance of the algorithm in the objective space.

### 2.3. Brain Storm Optimization

Shi [25] proposed the brain storm optimization (BSO) in 2011. In the BSO, the population is composed by  $N$  individuals, and each individual simulated the idea generated in the brainstorming process. Then the population is updated through the convergence and divergence operators. As a convergence operation, the clustering method is employed to divide the population. Then, the BSO can effectively improve the population diversity by selecting different individuals from a single cluster or multiple clusters for mutation.

The specific steps are as described below:

Step 1: Randomly generate  $N$  initial individuals and evaluate them.

Step 2: The initial population is clustered into multiple groups by the K-means method. Moreover, for each cluster, individuals are sorted and the best one is recorded as the cluster center.

Step 3: Randomly select a cluster center and generate a new individual to replace it.

Step 4: The new individual is produced as follows [25]:

If  $rand < p_1$  then

$$x_u^t = \begin{cases} x_{c1}, & \text{if } p_2 < 0.4 \\ x_{r1}, & \text{otherwise} \end{cases} \tag{4}$$

Else

$$x_u^t = \begin{cases} c * x_{c1} + (1 - c) * x_{c2}, & \text{if } p_3 < 0.5 \\ c * x_{r1} + (1 - c) * x_{r2}, & \text{otherwise} \end{cases} \tag{5}$$

End

where  $u$  is generated from 1 to  $N$ ,  $x_u^t$  is the  $u$ -th selected individual in the  $t$ -th iteration;  $x_{c1}$  and  $x_{c2}$  are random cluster centers;  $x_{r1}$  and  $x_{r2}$  are random individuals from two stomatic clusters.  $p_1$  is the probability to choose the individual from one cluster [25].  $p_2, p_3$ , and  $c$  are random number between 0 and 1. When  $rand < p_1$ , Equation (4) is used to select the individual, otherwise, Equation (5) is employed. And then, for the selected individuals, the Gaussian mutation is employed to generate new individuals.

$$\begin{aligned} \mathbf{x}_u^{t+1} &= \mathbf{x}_u^t + \zeta \times N(\mu, \sigma), \\ \zeta &= \log \operatorname{sig}\left(\frac{0.5 \times T - t}{z}\right) \times \operatorname{rand}(0, 1), \end{aligned} \quad (6)$$

where  $\mathbf{x}_u^{t+1}$  is the  $u$ -th new individual in the  $t + 1$ -th generation;  $N(\mu, \sigma)$  is the Gaussian distribution function with mean  $\mu$  and standard deviation  $\sigma$ .  $T$  represents the maximum number of iterations;  $\operatorname{rand}$  is a random value between 0 and 1;  $z$  is a slope.

Step 5: Evaluate new individuals and select the population for next generation.

Step 6: If the maximum number of iterations is reached, output the final solution, otherwise go to step 2.

### 3. Related Work

Finding the high-quality PF and preserving all equivalent Pareto solutions are two main tasks in the MMO. A large number of MMOEAs have been proposed to achieve the above objectives. For example, the Omni-optimizer proposed by Deb and Tiwari [26] introduced the crowding distance into the decision space. The results indicate that Pareto solutions with the same objective value can be effectively retained. Then, Liang et al. [4] designed a decision space-based niching NSGAI (DN-NSGAI). Experiments show that the proposed algorithm preserves almost all the PSs. In Ref. [23], a multi-objective PSO using ring topology and special crowding distance (SCD) (MO\_Ring\_PSO\_SCD) has been proposed. The ring topology is introduced to form stable neighborhoods. The SCD is a measurement method that comprehensively considers the density of the population. Their results exhibit that the proposed algorithm is effective in solving the MMOPs. In Ref. [12], a multi-objective PSO with a self-organizing map network (SMPSO-MM) is proposed. The self-organizing map network is utilized to establish multiple neighborhoods. The results expound that the proposed algorithm is superior to the compared algorithms. In Ref. [27], Zhang et al. proposed a two-stage search framework. The global search is applied in the first stage to identify as many optimal solutions as possible. Then, the DBSCN clustering with adaptive neighborhood radius is employed to enhance the local search ability. Their results demonstrate that the two-stage search framework is effective. In Ref. [28], the reference point strategy is used to construct the neighborhood and the dominant radius of each Pareto front is utilized to distinguish whether the individual is a local optimal solution. From the results, the proposed algorithm can effectively locate the local solutions. At the same time, the evolutionary algorithm using the hierarchy sorting [29] is proposed to locate the global and local PSs. In order to ensure population diversity, an evaluation method for the local convergence of the population is introduced in the hierarchy sorting. The simulation results demonstrate that the proposed algorithm can maintain both the global and local PSs. To improve the distribution of Pareto optimal solutions, a niche backtracking search algorithm is proposed by Hu et al. [30]. In the proposed algorithm, a novel mutation strategy, which is based on affinity propagation clustering and an adaptive local search, is employed. The achieved results show that the solutions obtained by the proposed algorithm are uniformly distributed in the decision space. Hu et al. [31] adopted two parallel offspring generation mechanisms to improve the population diversity, while the reverse vector mutation strategy and niching local search scheme were used to balance convergence and diversity. From the experimental results, it can be found that the proposed method outperforms its competitors in most functions. In Ref. [32], a decomposition-based algorithm is proposed to address MMOPs, and a density-based estimation strategy is designed to estimate the number of PSs. And the mean-shift algorithm is used to partition the population. Their results expound that the proposed algorithm can find and maintain multiple PSs. To solve the problem of excessive convergence, Ref. [33] introduced the adaptive parameter control method to promote the convergence of population. The adaptive sub-population size is used to balance the convergence of the population in each region. The results exhibit that the proposed algorithm is effective in solving the MMOPs with multiple solution sets. Ming et al. [34] proposed a co-evolutionary algorithm, in which the convergence-relaxed population is used to assist the convergence-first population in locating more PSs, and

the objective relaxation method is employed to complement previously undetected areas. The results validate the performance of the proposed algorithm is better than that of six advanced MMOEAs.

Most of the above-mentioned studies adopted the niching approaches to solve the MMOPs. Besides the “soft isolation” methods, “hard isolation” methods have been also proposed. In Ref. [13], Fan and Yan proposed the ZS strategy to maintain the population diversity in the decision space. The effectiveness of the ZS is systematically evaluated on 11 test functions. In Ref. [35], in order to distinguish the potentials of each subspace, the reinforcement learning method is used to dynamically allocate computing resources to each subspace. From the results, it is an effective strategy to assist the ZS in solving the MMOPs. Ji et al. [36] proposed a ZS and transfer learning-based MMEAs, in which transfer learning is employed to realize the information sharing between similar subspaces. The simulation results indicate that the proposed strategy is competitive.

#### 4. Proposed Algorithm

How to locate and preserve the equivalent PSs in the decision space is important for the MMO. And reducing the complexity of MMOPs and improving the global search capability may be effective. To achieve the above objectives, the ZS-MMBSO is proposed. In the ZS-MMBSO, the ZS strategy [13] is employed to reduce the size of the search space, the improved BSO is utilized to improve the global search ability.

##### 4.1. Search Space Segmentation

To reduce the complexity of MMOPs, the ZS is utilized to divide the decision space into some sub-regions. As in the previous study [13], randomly selecting  $h(1 \leq h \leq n)$  decision variables, and then each of them is divided into  $e$  equal parts. Therefore, the number of subspaces is set to  $w = e^h$ .

##### 4.2. Novel Individual Generation Strategy

Although the BSO has good performance in multimodal optimization, a fixed search pattern may be difficult to adapt to different evolutionary stages. Therefore, a novel individual generation strategy is proposed.

In the novel individual generation strategy, the Gaussian mutation has a higher probability to be selected in the early stage, and the DE/current-to-best/1 is employed in the latter stage. The main reason is that the Gaussian mutation can effectively explore the entire decision space and quickly find the area where the optimal solution may exist, while the DE/current-to-best/1 has good exploitation ability and can promote population convergence.

The steps of the novel individual generation strategy are as follows:

If  $rand < p_1$  then

$$x_u^t = \begin{cases} x_{nd}, & \text{if } p_2 < 0.4 \\ x_{r1}, & \text{otherwise} \end{cases} \quad (7)$$

Else

$$x_u^t = \begin{cases} c * x_{c1} + (1 - c) * x_{nd}, & \text{if } p_3 < 0.5 \\ c * x_{r1} + (1 - c) * x_{r2}, & \text{otherwise} \end{cases} \quad (8)$$

End if

If  $rand < t/T$  then

$$x_u^{t+1} = x_u^t + \zeta \times N(\mu, \sigma),$$

Else

$$x_u^{t+1} = x_u^t + F * (x_{nd} - x_u^t) + F * (x_{n1} - x_{n2}) \quad (9)$$

End

where  $x_u^t$  is the  $u$ -th selected individual in the  $t$ -th iteration;  $x_{r1}$  and  $x_{r2}$  are random individuals from two different stomatic clusters;  $x_{c1}$  is a random cluster center;  $x_{nd}$  is a

non-dominated solution from the  $k$ -th cluster; Random individuals from the  $k$ -th cluster are denoted as  $x_{n1}$  and  $x_{n2}$ .  $p_1$  is a pre-set probability parameter that determines the individuals for mutation is from the single cluster or multiple clusters.  $p_2$ ,  $p_3$ , and  $c$  are random number between 0 and 1. The remaining parameters remain consistent with the original reference [25].  $F$  is the scale factor.

#### 4.3. Main Framework of the Proposed Algorithm

The main framework of the proposed ZS-MMBSO is shown in Algorithm 1. Line 1 is the search space segmentation strategy which is used to divide the decision space. For each subspace, a population POP is randomly initialized (line 3). Line 5 is the judgment of the algorithm termination criterion. In Line 6, the K-means clustering is performed on the population POP. For each cluster, the non-dominated\_scd\_sort method [23] is employed in lines 7 to 9. The non-dominated individuals in  $k$ -th cluster  $C_k$  are stored in  $ND_k$ , and the optimal individual is stored in the cluster center archive CA. In lines 10 to 12, it is a certain probability that a cluster center will be replaced by a new individual. The probability parameter is the same as the original BSO [25]. In lines 13 to 17, offspring individuals are generated by the novel individual generation strategy in Section 4.2. The individuals in the offspring population and the population POP are sorted by the non-dominated\_scd\_sort method (line 18). The top N individuals are selected for the next iteration. If the stopping criterion is not met, the procedures in Lines 5–19 will be run iteratively. Otherwise, the non-dominated individuals in each subspace  $S_d$  are recorded as  $PS_d$  and  $PF_d$  in line 20. At last, the final PS and PF are selected.

---

#### Algorithm 1. Framework of ZS-MMBSO

---

Input: the number of subspaces:  $w$ ; maximum number of iterations:  $T$ ; the number of the cluster:  $K$ ; cluster center archive: CA; the archive of the individuals in the  $k$ -th cluster:  $C_k$ ; the archive of the non-dominated individuals in the  $k$ -th cluster:  $ND_k$ ; the number of the individuals in  $C_k$ :  $|C_k|$

```

1: Search space segmentation via Section 4.1.
2: for each subspace  $S_d$  in  $S_1, S_2, \dots, S_w$  do
3:   Initialize the population POP.
4:    $t = 1$ 
5:   if  $t < T$  then
6:     Use the K-means method to divide POP into  $K$  subpopulations.
7:     for  $k = 1 \rightarrow K$  do
8:       The non-dominated_scd_sort method is used to sort individuals in the  $k$ -th
cluster  $C_k$ . The non-dominated individuals of  $C_k$  are stored in  $ND_k$  and the optimal individual in
the  $C_k$  is stored in CA.
9:     end for
10:    if  $\text{rand} < 0.2$  then
11:      Randomly select an individual from the CA, and it is replaced by the new
individual, which is randomly generated within the search space.
12:    end if
13:    for  $k = 1 \rightarrow K$  do
14:      for  $l = 1 \rightarrow |C_k|$  do
15:        Generate offspring individuals via a novel individual genera-tion strategy in
Section 4.2.
16:      end for
17:    end for
18:    Perform the non-dominated_scd_sort method to choose the population for
next iteration.
19:  end if
20:  Record the non-dominated individuals in  $S_d$  as  $PS_d$  and  $PF_d$ .
21: end for
22:  $PS = \text{Selection}(PS_1 \cup PS_2 \cup \dots \cup PS_w)$  and  $PF = \text{Selection}(PF_1 \cup PF_2 \cup \dots \cup PF_w)$ .
Output: PS and PF

```

---

## 5. Experimental Results and Analyses

To evaluate the performance of the ZS-MMBSO, it is compared with five famous MMOEAs, which are the SMPSO\_MM [12], the MMGPE [37], the MMODE\_CSCD [6], the MMODE\_ICD [38], and the ZS-MO\_Ring\_PSO\_SCD [13]. The IEEE CEC2019 [22] MMOPs benchmark suite is employed to assess the performance of these comparison algorithms. Additionally, the experimental results are analyzed by Wilcoxon's rank sum test [39] and Friedman's test [40], and the significance level is set to be 0.05. "+", "-", and "=" indicate that the ZS-MMBSO is better, worse, or similar to the comparison algorithm.

### 5.1. Experimental Settings

The parameter settings of the selected MMOEAs are recommended by their original references [6,12,13,37,38]. For all compared algorithms, the maximum number of function evaluations and the population size are set to 80,000 and 800, respectively. All algorithms are implemented by using MATLAB (R2021a) and run independently for 20 times on each test problem. Additionally, some parameter settings of the ZS-MMBSO are as follows:  $F$  is set to 0.5 to scale the difference vectors; the pre-set probability parameter  $p_1$  is equal to 0.8; the number of clusters in K-means clustering is set to 20.

### 5.2. Comparison with Other Algorithms

Tables 1 and 2 record the mean values and standard deviations of  $PSP$  and  $HV$ , respectively. The best results are highlighted in bold, and the results are statistically analyzed by Wilcoxon's rank sum test.

From Table 1, it is observed that the ZS-MMBSO significantly outperforms the SMPSO\_MM, the MMGPE, the MMODE\_CSCD, and the ZS-MO\_Ring\_PSO\_SCD in terms of  $PSP$  on all MMOPs. The main reason may be that the ZS can greatly reduce the problem complexity, and the BSO algorithm has good global search capability. Namely, the proposed algorithm can integrate the advantages of "soft isolation" and "hard isolation" to maintain the population diversity. Therefore, compared with other competitors, the ZS-MMBSO can preserve more equivalent Pareto optimal solutions. Compared to the MMODE\_ICD, the ZS-MMBSO wins 20 test functions. Clearly, the overall performance of the proposed algorithm is significantly better than that of the MMODE\_ICD. For the other two remaining MMOPs (i.e., the SYM\_PART simple and the Omni\_test), the MMODE\_ICD performs better. For the SYM\_PART simple, most PSs are distributed in the boundary regions of subspaces, thus it is difficult for the ZS-MMBSO to search equivalent Pareto optimal solutions. For the Omni\_test, the distribution of PSs in different subspaces is imbalanced, thus the proposed algorithm may waste computing resources. To visualize the performance comparison of all algorithms, the Friedman test rankings are illustrated in Figure 1 and the ZS-MMBSO owns the best ranking among all compared algorithms. The above statistical results indicate the superiority of the ZS-MMBSO in decision space.

**Table 1.** Mean and standard deviation values of *PSP* obtained by different algorithms.

	SMPSO_MM		MMGPE		MMODE_CSCD		MMODE_ICD		ZS-MO_Ring_PSO_SCD		ZS-MMBSO
MMF1	$8.62 \times 10^1 (3.67 \times 10^0)$	+	$6.72 \times 10^1 (2.90 \times 10^0)$	+	$7.81 \times 10^1 (3.90 \times 10^0)$	+	$7.12 \times 10^1 (4.75 \times 10^0)$	+	$9.46 \times 10^1 (4.63 \times 10^0)$	+	$1.99 \times 10^2 (1.25 \times 10^1)$
MMF2	$1.38 \times 10^2 (2.15 \times 10^1)$	+	$1.32 \times 10^2 (1.90 \times 10^1)$	+	$2.90 \times 10^2 (4.52 \times 10^1)$	+	$2.11 \times 10^2 (3.26 \times 10^1)$	+	$1.33 \times 10^2 (1.78 \times 10^1)$	+	$1.22 \times 10^3 (2.21 \times 10^2)$
MMF3	$1.87 \times 10^2 (1.42 \times 10^1)$	+	$1.86 \times 10^2 (1.99 \times 10^1)$	+	$3.61 \times 10^2 (4.39 \times 10^1)$	+	$2.58 \times 10^2 (3.46 \times 10^1)$	+	$1.33 \times 10^2 (2.09 \times 10^1)$	+	$9.28 \times 10^2 (1.48 \times 10^2)$
MMF4	$1.39 \times 10^2 (4.21 \times 10^0)$	+	$1.11 \times 10^2 (4.40 \times 10^0)$	+	$1.49 \times 10^2 (4.72 \times 10^0)$	+	$1.57 \times 10^2 (6.50 \times 10^0)$	+	$1.90 \times 10^2 (5.97 \times 10^0)$	+	$3.64 \times 10^2 (1.10 \times 10^1)$
MMF5	$3.97 \times 10^1 (1.92 \times 10^0)$	+	$3.34 \times 10^1 (1.21 \times 10^0)$	+	$3.85 \times 10^1 (1.16 \times 10^0)$	+	$2.94 \times 10^1 (1.49 \times 10^0)$	+	$4.86 \times 10^1 (2.50 \times 10^0)$	+	$8.24 \times 10^1 (5.29 \times 10^0)$
MMF6	$4.23 \times 10^1 (1.48 \times 10^0)$	+	$3.59 \times 10^1 (9.56 \times 10^{-1})$	+	$4.24 \times 10^1 (1.16 \times 10^0)$	+	$3.37 \times 10^1 (1.38 \times 10^0)$	+	$5.45 \times 10^1 (2.95 \times 10^0)$	+	$8.04 \times 10^1 (4.25 \times 10^0)$
MMF7	$1.42 \times 10^2 (4.94 \times 10^0)$	+	$1.10 \times 10^2 (5.40 \times 10^0)$	+	$1.41 \times 10^2 (5.95 \times 10^0)$	+	$1.60 \times 10^2 (6.09 \times 10^0)$	+	$1.81 \times 10^2 (9.23 \times 10^0)$	+	$2.86 \times 10^2 (1.19 \times 10^1)$
MMF8	$6.25 \times 10^1 (3.47 \times 10^0)$	+	$4.26 \times 10^1 (2.60 \times 10^0)$	+	$6.30 \times 10^1 (2.28 \times 10^0)$	+	$2.66 \times 10^1 (6.91 \times 10^0)$	+	$6.40 \times 10^1 (2.73 \times 10^0)$	+	$1.67 \times 10^2 (2.03 \times 10^1)$
MMF9	$5.49 \times 10^2 (2.10 \times 10^1)$	+	$3.89 \times 10^2 (1.34 \times 10^1)$	+	$5.29 \times 10^2 (1.85 \times 10^1)$	+	$8.15 \times 10^2 (4.04 \times 10^1)$	+	$6.14 \times 10^2 (2.04 \times 10^1)$	+	$1.53 \times 10^3 (5.89 \times 10^1)$
MMF10	$2.55 \times 10^3 (1.89 \times 10^2)$	+	$1.77 \times 10^2 (3.15 \times 10^1)$	+	$1.98 \times 10^3 (1.80 \times 10^2)$	+	$1.41 \times 10^3 (4.65 \times 10^2)$	+	$1.37 \times 10^2 (1.59 \times 10^1)$	+	$3.16 \times 10^3 (2.91 \times 10^1)$
MMF11	$7.63 \times 10^2 (4.11 \times 10^1)$	+	$4.92 \times 10^2 (1.70 \times 10^1)$	+	$6.52 \times 10^2 (4.16 \times 10^1)$	+	$7.22 \times 10^2 (1.74 \times 10^1)$	+	$6.17 \times 10^2 (2.30 \times 10^1)$	+	$1.15 \times 10^3 (4.77 \times 10^1)$
MMF12	$1.29 \times 10^3 (9.39 \times 10^1)$	+	$6.17 \times 10^2 (5.65 \times 10^1)$	+	$1.46 \times 10^3 (5.47 \times 10^1)$	+	$1.01 \times 10^3 (2.84 \times 10^1)$	+	$8.68 \times 10^2 (4.59 \times 10^1)$	+	$1.83 \times 10^3 (6.82 \times 10^1)$
MMF13	$6.67 \times 10^1 (2.22 \times 10^0)$	+	$5.06 \times 10^1 (9.91 \times 10^{-1})$	+	$7.12 \times 10^1 (1.29 \times 10^0)$	+	$6.50 \times 10^1 (1.87 \times 10^0)$	+	$5.74 \times 10^1 (1.75 \times 10^0)$	+	$7.76 \times 10^1 (3.84 \times 10^0)$
MMF14	$2.98 \times 10^1 (4.26 \times 10^{-1})$	+	$3.00 \times 10^1 (4.25 \times 10^{-1})$	+	$3.15 \times 10^1 (8.69 \times 10^{-1})$	+	$4.00 \times 10^1 (5.84 \times 10^{-1})$	+	$5.32 \times 10^1 (5.36 \times 10^{-1})$	+	$5.77 \times 10^1 (8.39 \times 10^{-1})$
MMF15	$3.95 \times 10^1 (9.86 \times 10^{-1})$	+	$3.81 \times 10^1 (1.34 \times 10^0)$	+	$3.98 \times 10^1 (1.15 \times 10^0)$	+	$5.49 \times 10^1 (1.57 \times 10^0)$	+	$6.54 \times 10^1 (1.20 \times 10^0)$	+	$7.76 \times 10^1 (1.27 \times 10^0)$
MMF1_z	$1.19 \times 10^2 (5.12 \times 10^0)$	+	$9.01 \times 10^1 (2.91 \times 10^0)$	+	$1.13 \times 10^2 (5.02 \times 10^0)$	+	$5.52 \times 10^1 (4.38 \times 10^0)$	+	$1.27 \times 10^2 (4.47 \times 10^0)$	+	$2.70 \times 10^2 (1.90 \times 10^1)$
MMF1_e	$6.77 \times 10^0 (8.28 \times 10^{-1})$	+	$5.86 \times 10^0 (6.15 \times 10^{-1})$	+	$6.77 \times 10^0 (6.20 \times 10^{-1})$	+	$1.24 \times 10^0 (5.20 \times 10^{-1})$	+	$4.63 \times 10^0 (1.11 \times 10^0)$	+	$8.23 \times 10^0 (6.20 \times 10^{-1})$
MMF14_a	$2.66 \times 10^1 (5.83 \times 10^{-1})$	+	$2.66 \times 10^1 (5.65 \times 10^{-1})$	+	$2.62 \times 10^1 (4.20 \times 10^{-1})$	+	$2.90 \times 10^1 (4.97 \times 10^{-1})$	+	$4.82 \times 10^1 (4.79 \times 10^{-1})$	+	$4.94 \times 10^1 (9.01 \times 10^{-1})$
MMF15_a	$3.59 \times 10^1 (5.74 \times 10^{-1})$	+	$3.61 \times 10^1 (6.76 \times 10^{-1})$	+	$3.47 \times 10^1 (1.12 \times 10^0)$	+	$4.39 \times 10^1 (9.36 \times 10^{-1})$	+	$6.07 \times 10^1 (9.83 \times 10^{-1})$	+	$6.80 \times 10^1 (1.40 \times 10^0)$
SYM_PART simple	$3.15 \times 10^1 (2.52 \times 10^0)$	+	$1.23 \times 10^1 (1.27 \times 10^0)$	+	$6.17 \times 10^1 (1.68 \times 10^0)$	+	<b><math>1.03 \times 10^2 (4.45 \times 10^0)</math></b>	-	$2.16 \times 10^1 (9.85 \times 10^{-1})$	+	$9.22 \times 10^1 (9.45 \times 10^0)$
SYM_PART rotated	$1.67 \times 10^1 (1.08 \times 10^0)$	+	$2.40 \times 10^1 (1.55 \times 10^0)$	+	$6.27 \times 10^1 (2.57 \times 10^0)$	+	$5.98 \times 10^1 (3.06 \times 10^0)$	+	$2.03 \times 10^1 (1.00 \times 10^0)$	+	<b><math>1.25 \times 10^2 (1.45 \times 10^1)</math></b>
Omni_test	$7.64 \times 10^0 (1.50 \times 10^0)$	+	$5.10 \times 10^1 (1.05 \times 10^0)$	+	$8.16 \times 10^0 (3.18 \times 10^0)$	+	<b><math>5.46 \times 10^1 (1.97 \times 10^0)</math></b>	-	$9.34 \times 10^0 (1.29 \times 10^0)$	+	$3.04 \times 10^1 (4.22 \times 10^0)$
+		22		22		22		20		22	
=		0		0		0		0		0	
-		0		0		0		2		0	

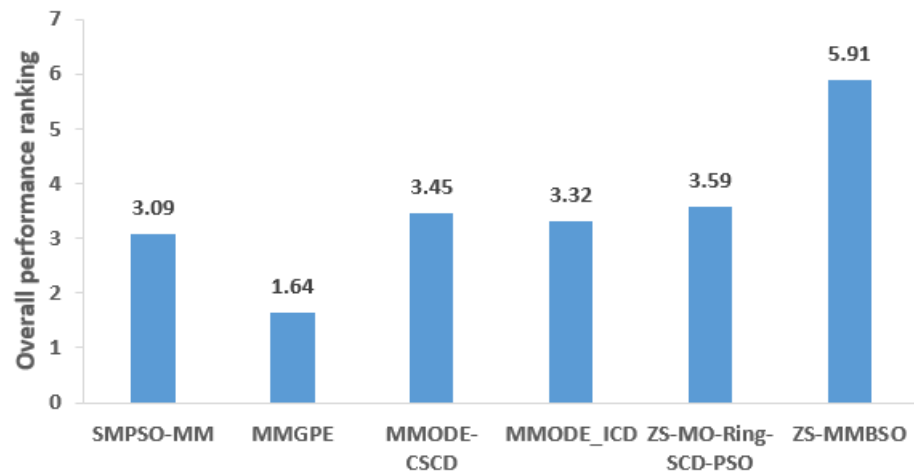
**Table 2.** Mean and standard deviation values of *HV* obtained by different algorithms.

	SMPSO_MM		MMGPE		MMODE_CSCD		MMODE_ICD		ZS-MO_Ring_PSO_SCD		ZS-MMBSO
MMF1	$8.76 \times 10^{-1} (5.23 \times 10^{-5})$	+	$8.75 \times 10^{-1} (6.88 \times 10^{-5})$	+	$8.76 \times 10^{-1} (3.64 \times 10^{-5})$	+	$8.76 \times 10^{-1} (1.13 \times 10^{-5})$	+	$8.76 \times 10^{-1} (6.69 \times 10^{-5})$	+	$8.76 \times 10^{-1} (3.63 \times 10^{-5})$
MMF2	$8.71 \times 10^{-1} (4.56 \times 10^{-4})$	+	$8.71 \times 10^{-1} (5.12 \times 10^{-4})$	+	$8.74 \times 10^{-1} (1.89 \times 10^{-4})$	+	$8.74 \times 10^{-1} (5.64 \times 10^{-4})$	+	$8.70 \times 10^{-1} (7.74 \times 10^{-4})$	+	$8.76 \times 10^{-1} (3.54 \times 10^{-4})$
MMF3	$8.72 \times 10^{-1} (4.01 \times 10^{-4})$	+	$8.72 \times 10^{-1} (3.94 \times 10^{-4})$	+	$8.74 \times 10^{-1} (9.24 \times 10^{-5})$	+	$8.75 \times 10^{-1} (2.50 \times 10^{-4})$	+	$8.70 \times 10^{-1} (9.45 \times 10^{-4})$	+	$8.76 \times 10^{-1} (2.50 \times 10^{-4})$
MMF4	$5.42 \times 10^{-1} (8.93 \times 10^{-5})$	+	$5.42 \times 10^{-1} (7.81 \times 10^{-5})$	+	$5.42 \times 10^{-1} (8.17 \times 10^{-5})$	+	$5.43 \times 10^{-1} (2.04 \times 10^{-5})$	+	$5.42 \times 10^{-1} (1.42 \times 10^{-4})$	+	$5.43 \times 10^{-1} (6.03 \times 10^{-5})$
MMF5	$8.76 \times 10^{-1} (3.66 \times 10^{-5})$	+	$8.75 \times 10^{-1} (5.18 \times 10^{-5})$	+	$8.76 \times 10^{-1} (4.94 \times 10^{-5})$	+	$8.76 \times 10^{-1} (1.92 \times 10^{-5})$	+	$8.76 \times 10^{-1} (6.58 \times 10^{-5})$	+	$8.76 \times 10^{-1} (5.03 \times 10^{-5})$
MMF6	$8.76 \times 10^{-1} (5.06 \times 10^{-5})$	+	$8.75 \times 10^{-1} (8.26 \times 10^{-5})$	+	$8.76 \times 10^{-1} (4.19 \times 10^{-5})$	+	$8.76 \times 10^{-1} (3.96 \times 10^{-5})$	+	$8.76 \times 10^{-1} (6.01 \times 10^{-5})$	+	$8.76 \times 10^{-1} (7.58 \times 10^{-5})$
MMF7	$8.76 \times 10^{-1} (2.79 \times 10^{-5})$	+	$8.75 \times 10^{-1} (9.18 \times 10^{-5})$	+	$8.76 \times 10^{-1} (2.53 \times 10^{-5})$	+	$8.76 \times 10^{-1} (2.03 \times 10^{-5})$	+	$8.76 \times 10^{-1} (3.04 \times 10^{-5})$	+	$8.76 \times 10^{-1} (2.34 \times 10^{-5})$



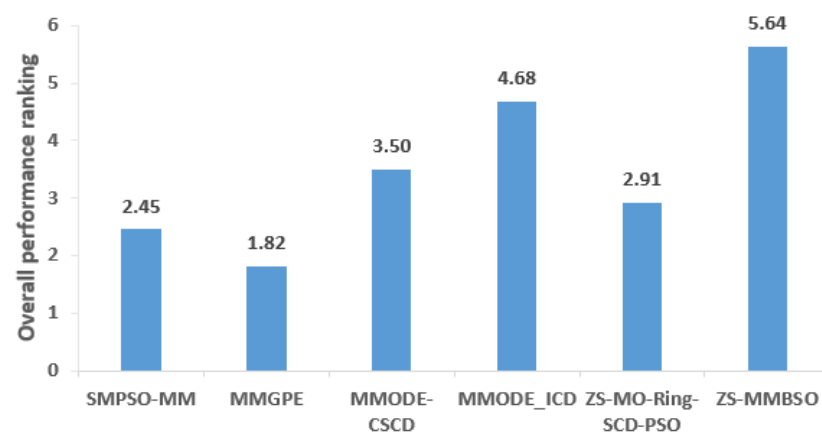
Table 2. Cont.

	SMP SO_MM		MMGPE		M MODE_CS CD		M MODE_IC D		Z S-M O_R ing_P SO_S CD		Z S-M MBS O
MMF8	$4.23 \times 10^{-1}$ ( $1.18 \times 10^{-4}$ )	+	$4.23 \times 10^{-1}$ ( $1.73 \times 10^{-4}$ )	+	$4.24 \times 10^{-1}$ ( $1.17 \times 10^{-4}$ )	+	$4.24 \times 10^{-1}$ ( $1.11 \times 10^{-4}$ )	+	$4.23 \times 10^{-1}$ ( $2.58 \times 10^{-4}$ )	+	<b><math>4.24 \times 10^{-1}</math> (<math>3.51 \times 10^{-4}</math>)</b>
MMF9	$9.70 \times 10^0$ ( $3.48 \times 10^{-4}$ )	+	$9.70 \times 10^0$ ( $5.97 \times 10^{-4}$ )	+	$9.70 \times 10^0$ ( $1.35 \times 10^{-4}$ )	+	$9.71 \times 10^0$ ( $3.01 \times 10^{-4}$ )	+	$9.71 \times 10^0$ ( $2.58 \times 10^{-4}$ )	+	<b><math>9.71 \times 10^0</math> (<math>3.15 \times 10^{-4}</math>)</b>
MMF10	$1.29 \times 10^1$ ( $9.73 \times 10^{-4}$ )	=	$1.43 \times 10^0$ ( $2.01 \times 10^{-2}$ )	+	<b><math>1.29 \times 10^1</math> (<math>1.62 \times 10^{-4}</math>)</b>	-	$1.28 \times 10^1$ ( $9.44 \times 10^{-2}$ )	=	$1.28 \times 10^1$ ( $1.09 \times 10^{-2}$ )	+	<b><math>1.29 \times 10^1</math> (<math>4.51 \times 10^{-3}</math>)</b>
MMF11	$1.45 \times 10^1$ ( $4.96 \times 10^{-4}$ )	+	$1.45 \times 10^1$ ( $5.56 \times 10^{-4}$ )	+	$1.45 \times 10^1$ ( $2.30 \times 10^{-4}$ )	+	$1.45 \times 10^1$ ( $2.76 \times 10^{-4}$ )	+	$1.45 \times 10^1$ ( $4.99 \times 10^{-4}$ )	+	<b><math>1.45 \times 10^1</math> (<math>4.48 \times 10^{-4}</math>)</b>
MMF12	$1.57 \times 10^0$ ( $1.32 \times 10^{-4}$ )	+	$1.57 \times 10^0$ ( $2.98 \times 10^{-4}$ )	+	$1.57 \times 10^0$ ( $6.55 \times 10^{-6}$ )	=	<b><math>1.57 \times 10^0</math> (<math>4.87 \times 10^{-5}</math>)</b>	=	$1.57 \times 10^0$ ( $6.83 \times 10^{-4}$ )	+	$1.57 \times 10^0$ ( $1.07 \times 10^{-4}$ )
MMF13	$1.85 \times 10^1$ ( $1.10 \times 10^{-3}$ )	+	$1.84 \times 10^1$ ( $2.30 \times 10^{-3}$ )	+	$1.85 \times 10^1$ ( $1.82 \times 10^{-4}$ )	+	$1.85 \times 10^1$ ( $6.12 \times 10^{-4}$ )	+	$1.84 \times 10^1$ ( $1.90 \times 10^{-3}$ )	+	<b><math>1.85 \times 10^1</math> (<math>6.77 \times 10^{-4}</math>)</b>
MMF14	$2.95 \times 10^0$ ( $9.22 \times 10^{-2}$ )	+	$3.02 \times 10^0$ ( $3.85 \times 10^{-2}$ )	+	$2.88 \times 10^0$ ( $1.31 \times 10^{-1}$ )	+	<b><math>3.16 \times 10^0</math> (<math>5.40 \times 10^{-2}</math>)</b>	=	$3.05 \times 10^0$ ( $2.12 \times 10^{-1}$ )	+	$3.15 \times 10^0$ ( $1.35 \times 10^{-1}$ )
MMF15	$4.33 \times 10^0$ ( $8.93 \times 10^{-2}$ )	+	$4.36 \times 10^0$ ( $3.04 \times 10^{-2}$ )	+	$4.13 \times 10^0$ ( $1.10 \times 10^{-1}$ )	+	<b><math>4.52 \times 10^0</math> (<math>5.16 \times 10^{-2}</math>)</b>	=	$4.40 \times 10^0$ ( $1.14 \times 10^{-1}$ )	+	$4.48 \times 10^0$ ( $1.13 \times 10^{-1}$ )
MMF1_z	$8.76 \times 10^{-1}$ ( $3.70 \times 10^{-5}$ )	+	$8.75 \times 10^{-1}$ ( $7.85 \times 10^{-5}$ )	+	$8.76 \times 10^{-1}$ ( $5.82 \times 10^{-5}$ )	+	$8.76 \times 10^{-1}$ ( $2.63 \times 10^{-5}$ )	+	$8.76 \times 10^{-1}$ ( $7.05 \times 10^{-5}$ )	+	<b><math>8.76 \times 10^{-1}</math> (<math>1.86 \times 10^{-5}</math>)</b>
MMF1_e	$8.73 \times 10^{-1}$ ( $8.39 \times 10^{-4}$ )	+	$8.71 \times 10^{-1}$ ( $9.87 \times 10^{-4}$ )	+	$8.64 \times 10^{-1}$ ( $5.10 \times 10^{-3}$ )	+	<b><math>8.75 \times 10^{-1}</math> (<math>2.49 \times 10^{-4}</math>)</b>	-	$8.70 \times 10^{-1}$ ( $9.00 \times 10^{-4}$ )	+	$8.74 \times 10^{-1}$ ( $8.20 \times 10^{-4}$ )
MMF14_a	$3.03 \times 10^0$ ( $6.85 \times 10^{-2}$ )	+	$3.05 \times 10^0$ ( $6.18 \times 10^{-2}$ )	+	$2.88 \times 10^0$ ( $1.72 \times 10^{-1}$ )	+	$3.12 \times 10^0$ ( $4.22 \times 10^{-2}$ )	+	$3.07 \times 10^0$ ( $1.49 \times 10^{-1}$ )	+	<b><math>3.23 \times 10^0</math> (<math>2.01 \times 10^{-1}</math>)</b>
MMF15_a	$4.27 \times 10^0$ ( $1.11 \times 10^{-1}$ )	+	$4.42 \times 10^0$ ( $4.35 \times 10^{-2}$ )	+	$4.16 \times 10^0$ ( $1.47 \times 10^{-1}$ )	+	$4.45 \times 10^0$ ( $4.24 \times 10^{-2}$ )	+	$4.47 \times 10^0$ ( $7.66 \times 10^{-2}$ )	=	<b><math>4.49 \times 10^0</math> (<math>7.99 \times 10^{-2}</math>)</b>
SYM_PART simple	$1.67 \times 10^1$ ( $1.20 \times 10^{-3}$ )	+	$1.66 \times 10^1$ ( $6.62 \times 10^{-3}$ )	+	$1.67 \times 10^1$ ( $1.14 \times 10^{-4}$ )	+	$1.67 \times 10^1$ ( $1.40 \times 10^{-4}$ )	+	$1.67 \times 10^1$ ( $1.50 \times 10^{-3}$ )	+	<b><math>1.67 \times 10^1</math> (<math>3.44 \times 10^{-5}</math>)</b>
SYM_PART rotated	$1.66 \times 10^1$ ( $2.20 \times 10^{-3}$ )	+	$1.67 \times 10^1$ ( $2.15 \times 10^{-3}$ )	+	$1.67 \times 10^1$ ( $2.06 \times 10^{-4}$ )	+	$1.67 \times 10^1$ ( $1.38 \times 10^{-4}$ )	+	$1.67 \times 10^1$ ( $2.00 \times 10^{-3}$ )	+	<b><math>1.67 \times 10^1</math> (<math>6.33 \times 10^{-5}</math>)</b>
Omni_test	$5.27 \times 10^1$ ( $1.00 \times 10^{-2}$ )	+	$5.27 \times 10^1$ ( $1.47 \times 10^{-2}$ )	+	$5.28 \times 10^1$ ( $6.06 \times 10^{-3}$ )	+	$5.28 \times 10^1$ ( $1.72 \times 10^{-3}$ )	+	$5.27 \times 10^1$ ( $1.28 \times 10^{-2}$ )	+	<b><math>5.28 \times 10^1</math> (<math>1.26 \times 10^{-3}</math>)</b>
+			21		22		20		17		21
=			1		0		1		4		1
-			0		0		1		1		0



**Figure 1.** Performance rankings of all compared algorithms in terms of *PSP* values.

Table 2 lists the *HV* results for these comparison algorithms. It is observed that the ZS-MMBSO outperforms the MMGPE on all test functions in terms of *HV*. Compared with the ZS-MO\_Ring\_PSO\_SCD and SMPSO\_MM, the proposed algorithm achieves better results on 21 test functions and similar result on one test function. The ZS-MMBSO is superior to the MMODE\_CSCD on 20 test functions. For the MMF10, the proposed algorithm is surpassed by the MMODE\_CSCD. The reason is that the global PS of MMF10 is simple, and it is easier for MMODE\_CSCD to find the PF approximation. The MMODE\_ICD performs better than the ZS-MMBSO on the MMF1\_e, but their difference is small. In general, the proposed algorithm is competitive in the objective space. One of the reasons is that the ZS-MMBSO integrates the “soft isolation” and the “hard isolation” to find more PSs, which the population diversity in both decision space and objective space will be improved. The novel individual generation strategy can achieve a trade-off between global exploration and local exploitation, which can balance diversity and convergence. However, the performance improvement of the proposed algorithm in the objective space is not more significant than that in the decision space. In the MMOPs, there are multiple equivalent solutions in the decision space that are corresponding to the same objective value. It is easier for MMOEAs to find a good PF approximation than to locate equivalent solutions in the decision space. As a result, the performance difference of the comparison algorithms in the objective space is relatively small. Figure 2 shows Friedman’s test results of the overall performance ranking. The ZS-MMBSO gets the first ranking, followed by MMODE\_ICD, MMODE\_CSCD, ZS-MO\_Ring\_PSO\_SCD, SMPSO\_MM, and MMGPE. The above observations fully indicate the ZS-MMBSO performs better in the objective space.



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

Based on the above analyses, the ZS-MMBSO performs the best among all competitors in both decision space and objective space. The proposed algorithm is an effective tool to solve different types of MMOPs.

### 5.3. Experimental Analysis

#### 5.3.1. The Effectiveness of the ZS

To validate the effectiveness of the ZS, the ZS-MMBSO and ZS-MMBSO without the ZS strategy (denoted as ZS-MMBSO\_1) are used to compare 22 functions in terms of *PSP*. The parameter settings are consistent with Section 5.1. For each function, the optimal value is displayed in bold, and the results are analyzed by Wilcoxon’s rank sum test.

In Table 3, the mean *PSP* values of the ZS-MMBSO and its variant are presented. It can be found that ZS-MMBSO performs better than ZS-MMBSO\_1 on all test functions. So, the search space segmentation has a significant impact on the proposed algorithm in the decision space. It is because the search complexity is reduced by the ZS, this is important for solving complex problems. In summary, the ZS is effective to assist the proposed algorithm in locating more equivalent PSs.

**Table 3.** Mean and standard deviation values of *PSP* obtained by ZS-MMBSO and ZS-MMBSO\_1.

	ZS-MMBSO_1		ZS-MMBSO
MMF1	$8.38 \times 10^1$ ( $2.95 \times 10^0$ )	+	<b><math>1.99 \times 10^2</math></b> ( <b><math>1.25 \times 10^1</math></b> )
MMF2	$3.90 \times 10^2$ ( $9.44 \times 10^1$ )	+	<b><math>1.22 \times 10^3</math></b> ( <b><math>2.21 \times 10^2</math></b> )
MMF3	$2.94 \times 10^2$ ( $7.88 \times 10^1$ )	+	<b><math>9.28 \times 10^2</math></b> ( <b><math>1.48 \times 10^2</math></b> )
MMF4	$8.42 \times 10^1$ ( $1.25 \times 10^1$ )	+	<b><math>3.64 \times 10^2</math></b> ( <b><math>1.10 \times 10^1</math></b> )
MMF5	$3.36 \times 10^1$ ( $2.08 \times 10^0$ )	+	<b><math>8.24 \times 10^1</math></b> ( <b><math>5.29 \times 10^0</math></b> )
MMF6	$3.75 \times 10^1$ ( $1.65 \times 10^0$ )	+	<b><math>8.04 \times 10^1</math></b> ( <b><math>4.25 \times 10^0</math></b> )
MMF7	$1.39 \times 10^2$ ( $6.75 \times 10^0$ )	+	<b><math>2.86 \times 10^2</math></b> ( <b><math>1.19 \times 10^0</math></b> )
MMF8	$2.66 \times 10^1$ ( $7.53 \times 10^0$ )	+	<b><math>1.67 \times 10^2</math></b> ( <b><math>2.03 \times 10^1</math></b> )
MMF9	$5.30 \times 10^2$ ( $1.02 \times 10^2$ )	+	<b><math>1.53 \times 10^3</math></b> ( <b><math>5.89 \times 10^1</math></b> )
MMF10	$2.34 \times 10^3$ ( $1.53 \times 10^1$ )	+	<b><math>3.16 \times 10^3</math></b> ( <b><math>2.91 \times 10^1</math></b> )
MMF11	$7.98 \times 10^2$ ( $6.19 \times 10^1$ )	+	<b><math>1.15 \times 10^3</math></b> ( <b><math>4.77 \times 10^1</math></b> )
MMF12	$1.33 \times 10^3$ ( $2.23 \times 10^1$ )	+	<b><math>1.83 \times 10^3</math></b> ( <b><math>6.82 \times 10^1</math></b> )
MMF13	$4.36 \times 10^1$ ( $9.91 \times 10^0$ )	+	<b><math>7.76 \times 10^1</math></b> ( <b><math>3.84 \times 10^0</math></b> )
MMF14	$2.68 \times 10^1$ ( $1.92 \times 10^0$ )	+	<b><math>5.77 \times 10^1</math></b> ( <b><math>8.39 \times 10^{-1}</math></b> )
MMF15	$3.90 \times 10^1$ ( $1.20 \times 10^0$ )	+	<b><math>7.76 \times 10^1</math></b> ( <b><math>1.27 \times 10^0</math></b> )
MMF1_z	$9.62 \times 10^1$ ( $5.76 \times 10^0$ )	+	<b><math>2.70 \times 10^2</math></b> ( <b><math>1.90 \times 10^1</math></b> )
MMF1_e	$5.37 \times 10^0$ ( $1.59 \times 10^0$ )	+	<b><math>8.23 \times 10^0</math></b> ( <b><math>6.20 \times 10^{-1}</math></b> )
MMF14_a	$2.18 \times 10^1$ ( $1.23 \times 10^0$ )	+	<b><math>4.94 \times 10^1</math></b> ( <b><math>9.01 \times 10^{-1}</math></b> )
MMF15_a	$3.07 \times 10^1$ ( $1.96 \times 10^0$ )	+	<b><math>6.80 \times 10^1</math></b> ( <b><math>1.40 \times 10^0</math></b> )
SYM_PART simple	$3.70 \times 10^1$ ( $1.25 \times 10^1$ )	+	<b><math>9.22 \times 10^1</math></b> ( <b><math>9.45 \times 10^0</math></b> )
SYM_PART rotated	$6.30 \times 10^1$ ( $8.06 \times 10^0$ )	+	<b><math>1.25 \times 10^2</math></b> ( <b><math>1.45 \times 10^1</math></b> )
Omni_test	$1.98 \times 10^0$ ( $5.10 \times 10^{-1}$ )	+	<b><math>3.04 \times 10^1</math></b> ( <b><math>4.22 \times 10^0</math></b> )
		+	22
		=	0
		-	0

#### 5.3.2. The Effectiveness of the Novel Individual Generation Strategy

To demonstrate the effectiveness of the novel individual generation strategy, the *PSP* is used to testify the performance of the proposed algorithm and its variant. The ZS-MMBSO and ZS-MMBSO which adopts the Gaussian mutation to generate the offspring individuals (named as ZS-MMBSO\_2) are compared on 22 test functions. The parameter settings are shown in Section 5.1. The Wilcoxon’s rank sum test is used to analyze the results and the best results are showed in bold.

From Table 4, it can be observed that ZS-MMBSO achieves the best results in 19 out of 22 cases. On the MMF14\_a and the SYM\_PART simple, the ZS-MMBSO\_2 obtains the best results. The statistical results indicate the effectiveness of the novel individual generation

strategy. It is because the novel individual generation strategy can balance the global exploration and local exploitation. The Gaussian mutation used in the early stage can explore the entire search space and find the potential regions where the optimal solution may be located. By using the information of the non-dominated solutions, the DE/current-to-best/1 can guide individuals to search the potential regions and locate more equivalent solutions. The mutation strategy used in the latter stage has a good exploitation ability and can speed up the population convergence. For MMF14\_a and SYM\_PART simple, the ZS-MMBSO\_2 performs better than ZS-MMBSO, but the difference is small. Overall, the novel individual generation strategy can help the proposed algorithm locate the equivalent PSs in the decision space.

**Table 4.** Mean and standard deviation values of *PSP* obtained by ZS-MMBSO and ZS-MMBSO\_2.

	ZS-MMBSO_2		ZS-MMBSO
MMF1	$1.24 \times 10^2$ ( $4.88 \times 10^0$ )	+	$1.99 \times 10^2$ ( $1.25 \times 10^1$ )
MMF2	$1.69 \times 10^2$ ( $9.98 \times 10^0$ )	+	$1.22 \times 10^3$ ( $2.21 \times 10^2$ )
MMF3	$1.74 \times 10^2$ ( $1.34 \times 10^1$ )	+	$9.28 \times 10^2$ ( $1.48 \times 10^2$ )
MMF4	$2.64 \times 10^2$ ( $9.27 \times 10^0$ )	+	$3.64 \times 10^2$ ( $1.10 \times 10^1$ )
MMF5	$6.17 \times 10^1$ ( $2.54 \times 10^0$ )	+	$8.24 \times 10^1$ ( $5.29 \times 10^0$ )
MMF6	$7.52 \times 10^1$ ( $3.69 \times 10^0$ )	+	$8.04 \times 10^1$ ( $4.25 \times 10^0$ )
MMF7	$2.54 \times 10^2$ ( $1.13 \times 10^1$ )	+	$2.86 \times 10^2$ ( $1.19 \times 10^0$ )
MMF8	$1.21 \times 10^2$ ( $6.61 \times 10^0$ )	+	$1.67 \times 10^2$ ( $2.03 \times 10^1$ )
MMF9	$7.15 \times 10^2$ ( $1.94 \times 10^1$ )	+	$1.53 \times 10^3$ ( $5.89 \times 10^1$ )
MMF10	$2.30 \times 10^2$ ( $2.85 \times 10^1$ )	+	$3.16 \times 10^3$ ( $2.91 \times 10^1$ )
MMF11	$7.14 \times 10^2$ ( $2.91 \times 10^1$ )	+	$1.15 \times 10^3$ ( $4.77 \times 10^1$ )
MMF12	$7.08 \times 10^2$ ( $4.92 \times 10^1$ )	+	$1.83 \times 10^3$ ( $6.82 \times 10^1$ )
MMF13	$6.23 \times 10^1$ ( $2.15 \times 10^0$ )	+	$7.76 \times 10^1$ ( $3.84 \times 10^0$ )
MMF14	$5.77 \times 10^1$ ( $6.05 \times 10^{-1}$ )	=	$5.77 \times 10^1$ ( $8.39 \times 10^{-1}$ )
MMF15	$7.41 \times 10^1$ ( $9.13 \times 10^{-1}$ )	+	$7.76 \times 10^1$ ( $1.27 \times 10^0$ )
MMF1_z	$1.63 \times 10^2$ ( $5.58 \times 10^0$ )	+	$2.70 \times 10^2$ ( $1.90 \times 10^1$ )
MMF1_e	$6.76 \times 10^0$ ( $7.15 \times 10^{-1}$ )	+	$8.23 \times 10^0$ ( $6.20 \times 10^{-1}$ )
MMF14_a	$5.11 \times 10^1$ ( $5.41 \times 10^{-1}$ )	-	$4.94 \times 10^1$ ( $9.01 \times 10^{-1}$ )
MMF15_a	$6.68 \times 10^1$ ( $8.58 \times 10^{-1}$ )	+	$6.80 \times 10^1$ ( $1.40 \times 10^0$ )
SYM_PART simple	$1.00 \times 10^2$ ( $4.37 \times 10^0$ )	-	$9.22 \times 10^1$ ( $9.45 \times 10^0$ )
SYM_PART rotated	$1.01 \times 10^2$ ( $4.36 \times 10^0$ )	+	$1.25 \times 10^2$ ( $1.45 \times 10^1$ )
Omni_test	$1.99 \times 10^1$ ( $1.62 \times 10^0$ )	+	$3.04 \times 10^1$ ( $4.22 \times 10^0$ )
		+	19
		=	1
		-	2

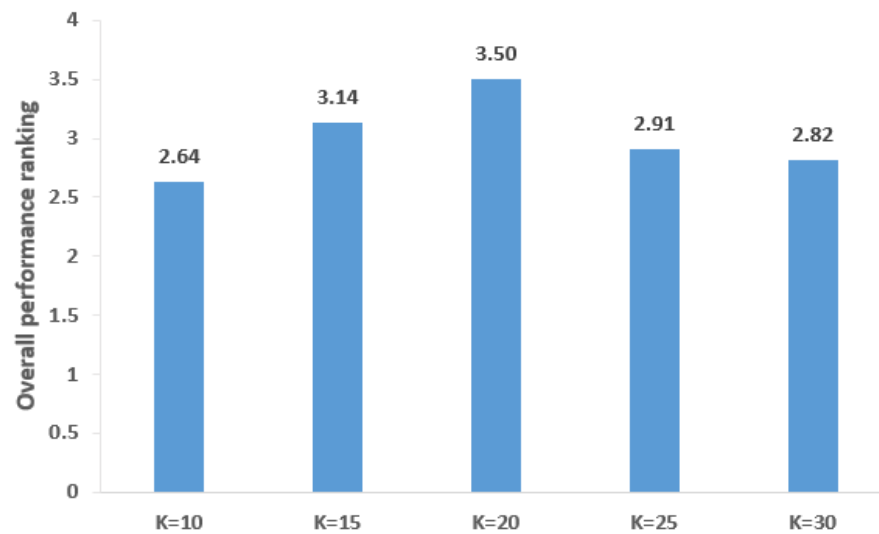
### 5.3.3. Impact of the Number of Clusters

The number of clusters may impact the performance of the BSO. If there are too many clusters, the population diversity will be improved. Conversely, if the number of clusters is too small, the population may converge quickly, and the equivalent solutions will be difficult to preserve. To validate the influence of different *K* values on the performance of the ZS-MMBSO, *K* is set to be 10, 15, 20, 25, and 30, respectively. In addition, ZS-MMBSO with different *K* values is independently run 20 times on 22 functions. The best results are marked in bold. And Friedman’s test [40] is employed to analyze the experimental results.

In Table 5, it is observed that for different *K* values, the ZS-MMBSO can only achieve the best results on a few problems. Namely, the differences for different *K* values are small. This means that users can easily set the *K* value. The Friedman rankings with different *K* values are represented in Figure 3. From Figure 3, if the number of clusters is too large or small, the performance of the proposed algorithm will be degraded. When *K* = 20, the ZS-MMBSO can balance diversity and convergence and its overall performance is the best. Hence, the *K* value of ZS-MMBSO is set to 20.

**Table 5.** Mean and standard deviation values of *PSP* obtained by different *K* values.

	10	15	20	25	30
MMF1	$1.66 \times 10^2$ ( $9.23 \times 10^0$ )	$1.76 \times 10^2$ ( $9.95 \times 10^{-1}$ )	$1.99 \times 10^2$ ( $1.25 \times 10^1$ )	$2.07 \times 10^2$ ( $1.01 \times 10^1$ )	<b><math>2.16 \times 10^2</math> (<math>1.21 \times 10^1</math>)</b>
MMF2	$1.04 \times 10^3$ ( $1.42 \times 10^2$ )	$1.16 \times 10^3$ ( $1.24 \times 10^2$ )	$1.22 \times 10^3$ ( $2.21 \times 10^2$ )	<b><math>1.31 \times 10^3</math> (<math>1.77 \times 10^2</math>)</b>	$1.25 \times 10^3$ ( $1.96 \times 10^2$ )
MMF3	$7.97 \times 10^2$ ( $1.64 \times 10^2$ )	<b><math>9.37 \times 10^2</math> (<math>1.20 \times 10^2</math>)</b>	$9.28 \times 10^2$ ( $1.48 \times 10^2$ )	$9.16 \times 10^2$ ( $2.04 \times 10^2$ )	$9.21 \times 10^2$ ( $1.54 \times 10^2$ )
MMF4	$3.52 \times 10^2$ ( $8.24 \times 10^0$ )	$3.59 \times 10^2$ ( $9.81 \times 10^0$ )	<b><math>3.64 \times 10^2</math> (<math>1.10 \times 10^1</math>)</b>	$3.57 \times 10^2$ ( $1.27 \times 10^1$ )	$3.56 \times 10^2$ ( $1.23 \times 10^1$ )
MMF5	$7.45 \times 10^1$ ( $4.39 \times 10^0$ )	$7.89 \times 10^1$ ( $4.28 \times 10^0$ )	$8.24 \times 10^1$ ( $5.29 \times 10^0$ )	$8.32 \times 10^1$ ( $3.96 \times 10^0$ )	<b><math>8.75 \times 10^1</math> (<math>4.28 \times 10^0</math>)</b>
MMF6	$7.68 \times 10^1$ ( $3.48 \times 10^0$ )	$7.86 \times 10^1$ ( $2.60 \times 10^0$ )	$8.04 \times 10^1$ ( $4.25 \times 10^0$ )	$8.29 \times 10^1$ ( $2.99 \times 10^0$ )	<b><math>8.36 \times 10^1</math> (<math>3.64 \times 10^0</math>)</b>
MMF7	$2.70 \times 10^2$ ( $1.20 \times 10^1$ )	$2.75 \times 10^2$ ( $1.19 \times 10^1$ )	$2.86 \times 10^2$ ( $1.19 \times 10^1$ )	$2.84 \times 10^2$ ( $1.26 \times 10^1$ )	<b><math>2.94 \times 10^2</math> (<math>1.04 \times 10^1</math>)</b>
MMF8	$1.66 \times 10^2$ ( $2.11 \times 10^1$ )	$1.67 \times 10^2$ ( $2.44 \times 10^1$ )	$1.67 \times 10^2$ ( $2.03 \times 10^1$ )	<b><math>1.68 \times 10^2</math> (<math>1.91 \times 10^1</math>)</b>	$1.65 \times 10^2$ ( $2.12 \times 10^1$ )
MMF9	<b><math>1.53 \times 10^3</math> (<math>6.41 \times 10^1</math>)</b>	$1.53 \times 10^3$ ( $6.10 \times 10^1$ )	$1.53 \times 10^3$ ( $5.89 \times 10^1$ )	$1.52 \times 10^3$ ( $5.70 \times 10^1$ )	$1.52 \times 10^3$ ( $5.47 \times 10^1$ )
MMF10	$2.97 \times 10^3$ ( $6.61 \times 10^2$ )	$3.08 \times 10^3$ ( $7.65 \times 10^2$ )	<b><math>3.16 \times 10^3</math> (<math>2.91 \times 10^1</math>)</b>	$2.84 \times 10^3$ ( $7.06 \times 10^2$ )	$2.65 \times 10^3$ ( $7.47 \times 10^2$ )
MMF11	<b><math>1.17 \times 10^3</math> (<math>5.05 \times 10^1</math>)</b>	$1.15 \times 10^3$ ( $4.82 \times 10^1$ )	$1.15 \times 10^3$ ( $4.77 \times 10^1$ )	$1.15 \times 10^3$ ( $5.45 \times 10^1$ )	$1.15 \times 10^3$ ( $4.33 \times 10^1$ )
MMF12	$1.83 \times 10^3$ ( $7.57 \times 10^1$ )	<b><math>1.84 \times 10^3</math> (<math>7.17 \times 10^1</math>)</b>	$1.83 \times 10^3$ ( $6.82 \times 10^1$ )	$1.83 \times 10^3$ ( $6.19 \times 10^1$ )	$1.76 \times 10^3$ ( $8.67 \times 10^1$ )
MMF13	$7.42 \times 10^1$ ( $6.60 \times 10^0$ )	$7.59 \times 10^1$ ( $6.25 \times 10^0$ )	<b><math>7.76 \times 10^1</math> (<math>3.84 \times 10^0</math>)</b>	$7.73 \times 10^1$ ( $4.06 \times 10^0$ )	$7.66 \times 10^1$ ( $3.86 \times 10^0$ )
MMF14	<b><math>5.86 \times 10^1</math> (<math>8.46 \times 10^{-1}</math>)</b>	$5.80 \times 10^1$ ( $5.65 \times 10^{-1}$ )	$5.77 \times 10^1$ ( $8.39 \times 10^{-1}$ )	$5.75 \times 10^1$ ( $8.70 \times 10^{-1}$ )	$5.68 \times 10^1$ ( $8.82 \times 10^{-1}$ )
MMF15	<b><math>7.80 \times 10^1</math> (<math>1.06 \times 10^0</math>)</b>	$7.77 \times 10^1$ ( $1.16 \times 10^0$ )	$7.76 \times 10^1$ ( $1.27 \times 10^0$ )	$7.63 \times 10^1$ ( $1.31 \times 10^0$ )	$7.62 \times 10^1$ ( $1.22 \times 10^0$ )
MMF1_z	$2.24 \times 10^2$ ( $2.22 \times 10^1$ )	$2.54 \times 10^2$ ( $1.71 \times 10^1$ )	$2.70 \times 10^2$ ( $1.90 \times 10^1$ )	$2.91 \times 10^2$ ( $1.74 \times 10^1$ )	<b><math>3.00 \times 10^2</math> (<math>1.66 \times 10^1</math>)</b>
MMF1_e	$7.17 \times 10^0$ ( $9.16 \times 10^{-1}$ )	$8.24 \times 10^0$ ( $7.13 \times 10^{-1}$ )	$8.23 \times 10^0$ ( $6.20 \times 10^{-1}$ )	$8.23 \times 10^0$ ( $1.03 \times 10^0$ )	<b><math>8.59 \times 10^0</math> (<math>9.81 \times 10^{-1}</math>)</b>
MMF14_a	$5.02 \times 10^1$ ( $7.29 \times 10^{-1}$ )	<b><math>4.96 \times 10^1</math> (<math>5.89 \times 10^{-1}</math>)</b>	$4.94 \times 10^1$ ( $9.01 \times 10^{-1}$ )	$4.86 \times 10^1$ ( $1.02 \times 10^0$ )	$4.84 \times 10^1$ ( $6.84 \times 10^{-1}$ )
MMF15_a	<b><math>6.93 \times 10^1</math> (<math>1.33 \times 10^0</math>)</b>	$6.85 \times 10^1$ ( $1.60 \times 10^0$ )	$6.80 \times 10^1$ ( $1.40 \times 10^0$ )	$6.72 \times 10^1$ ( $1.56 \times 10^0$ )	$6.67 \times 10^1$ ( $1.24 \times 10^0$ )
SYM_PART simple	<b><math>1.01 \times 10^2</math> (<math>1.10 \times 10^1</math>)</b>	$9.60 \times 10^1$ ( $1.01 \times 10^1$ )	$9.22 \times 10^1$ ( $9.45 \times 10^0$ )	$9.20 \times 10^1$ ( $9.48 \times 10^0$ )	$8.50 \times 10^1$ ( $9.85 \times 10^0$ )
SYM_PART rotated	$1.24 \times 10^2$ ( $1.51 \times 10^1$ )	$1.21 \times 10^2$ ( $1.90 \times 10^1$ )	$1.25 \times 10^2$ ( $1.45 \times 10^1$ )	$1.22 \times 10^2$ ( $1.68 \times 10^1$ )	<b><math>1.26 \times 10^2</math> (<math>1.12 \times 10^1</math>)</b>
Omni_test	$1.38 \times 10^1$ ( $6.81 \times 10^0$ )	<b><math>3.11 \times 10^1</math> (<math>8.24 \times 10^0</math>)</b>	$3.04 \times 10^1$ ( $4.22 \times 10^0$ )	$3.10 \times 10^1$ ( $3.01 \times 10^0$ )	$3.01 \times 10^1$ ( $5.28 \times 10^0$ )



**Figure 3.** Performance rankings of *PSP* values with different *K* values.

## 6. Discussion

In this study, a zoning search-based multimodal multi-objective brain storm optimization (ZS-MMBSO) is proposed. In the ZS-MMBSO, the ZS is employed to reduce the complexity of MMOPs. A novel individual generation strategy is proposed to strengthen the global exploration ability in the early stage and improve local exploitation in the late stage. To demonstrate the effectiveness of the ZS-MMBSO, it is compared with five famous MMOEAs on 22 test functions. The results demonstrate ZS-MMBSO is competitive in both decision space and objective space. The effectiveness of the proposed strategy is analyzed in Section 5.3.1 and Section 5.3.2. The experimental analysis results indicate that these strategies can help the ZS-MMBSO solve MMOPs. Section 5.3.3 analyzed the sensitivity of the cluster number *K*.

**Author Contributions:** Conceptualization, J.F. and Q.F.; methodology, J.F., W.H., Q.J. and Q.F.; investigation, J.F., W.H., Q.J. and Q.F.; writing—original draft preparation, J.F.; writing—review and editing, J.F., W.H., Q.J. and Q.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China-Shandong joint fund (U2006228), the National Natural Science Foundation of China (No. 61603244, No. 71904116), and the Shanghai Pujiang Program (No. 22PJJD030).

**Data Availability Statement:** Data supporting reported results are available from the authors upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
2. Hartikainen, M.; Miettinen, K.; Wiecek, M.M. PAINT: Pareto front interpolation for nonlinear multiobjective optimization. *Comput. Optim. Appl.* **2012**, *52*, 845–867. [[CrossRef](#)]
3. Nedjah, N.; Mourelle, L.D. Evolutionary multi-objective optimisation: A survey. *Int. J. Bio-Inspired Comput.* **2015**, *7*, 1–25. [[CrossRef](#)]
4. Liang, J.J.; Yue, C.T.; Qu, B.Y. Multimodal multi-objective optimization: A preliminary study. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2454–2461.
5. Li, G.; Wang, W.; Chen, H.; You, W.; Wang, Y.; Jin, Y.; Zhang, W. A SHADE-based multimodal multi-objective evolutionary algorithm with fitness sharing. *Appl. Intell.* **2021**, *51*, 8720–8752. [[CrossRef](#)]
6. Liang, J.; Qiao, K.; Yue, C.; Yu, K.; Qu, B.; Xu, R.; Li, Z.; Hu, Y. A clustering-based differential evolution algorithm for solving multimodal multi-objective optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100788. [[CrossRef](#)]

7. Qu, B.; Li, C.; Liang, J.; Yan, L.; Yu, K.; Zhu, Y. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems. *Appl. Soft Comput. J.* **2020**, *86*, 105886. [[CrossRef](#)]
8. Wang, W.; Li, G.; Wang, Y.; Wu, F.; Zhang, W.; Li, L. Clearing-based multimodal multi-objective evolutionary optimization with layer-to-layer strategy. *Swarm Evol. Comput.* **2022**, *68*, 100976. [[CrossRef](#)]
9. Li, W.; Zhang, T.; Wang, R.; Ishibuchi, H. Weighted Indicator-Based Evolutionary Algorithm for Multimodal Multiobjective Optimization. *IEEE Trans. Evol. Comput.* **2021**, *25*, 1064–1078. [[CrossRef](#)]
10. Li, G.; Wang, W.; Zhang, W.; Wang, Z.; Tu, H.; You, W. Grid search based multi-population particle swarm optimization algorithm for multimodal multi-objective optimization. *Swarm Evol. Comput.* **2021**, *62*, 100843. [[CrossRef](#)]
11. Qu, B.; Li, G.; Yan, L.; Liang, J.; Yue, C.; Yu, K.; Crisalle, O.D. A grid-guided particle swarm optimizer for multimodal multi-objective problems. *Appl. Soft Comput.* **2022**, *117*, 108381. [[CrossRef](#)]
12. Liang, J.; Guo, Q.; Yue, C.; Qu, B.; Yu, K. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In Proceedings of the 9th International Conference on Swarm Intelligence, ICSI 2018, Shanghai, China, 17–22 June 2018; Springer: Shanghai, China, 2018; pp. 550–560.
13. Fan, Q.; Yan, X. Solving Multimodal Multiobjective Problems Through Zoning Search. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 4836–4847. [[CrossRef](#)]
14. Fan, Q.; Ersoy, O.K. Zoning Search With Adaptive Resource Allocating Method for Balanced and Imbalanced Multimodal Multi-Objective Optimization. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 1163–1176. [[CrossRef](#)]
15. Guo, Y.-N.; Jiang, D.-Z.; Wang, R.-R.; Gong, D.-W. Structural design of heat exchanger plate with wide-channel based on multi-objective brain storm optimization. *Kongzhi Yu Juece/Control Decis.* **2022**, *37*, 2314–2322. [[CrossRef](#)]
16. Hou, Y.; Wang, H.; Fu, Y.; Gao, K.; Zhang, H. Multi-Objective brain storm optimization for integrated scheduling of distributed flow shop and distribution with maximal processing quality and minimal total weighted earliness and tardiness. *Comput. Ind. Eng.* **2023**, *179*, 109217. [[CrossRef](#)]
17. Cheng, S.; Chen, J.; Lei, X.; Shi, Y. Locating Multiple Optima via Brain Storm Optimization Algorithms. *IEEE Access* **2018**, *6*, 17039–17049. [[CrossRef](#)]
18. Dai, Z.; Fang, W.; Li, Q.; Chen, W.-N. Modified Self-adaptive Brain Storm Optimization Algorithm for Multimodal Optimization. In Proceedings of the 14th International Conference on Bio-inspired Computing: Theories and Applications, BIC-TA 2019, Zhengzhou, China, 22–25 November 2019; Springer Science and Business Media Deutschland GmbH: Zhengzhou, China, 2020; pp. 384–397.
19. Dai, Z.; Fang, W.; Tang, K.; Li, Q. An optima-identified framework with brain storm optimization for multimodal optimization problems. *Swarm Evol. Comput.* **2021**, *62*, 100827. [[CrossRef](#)]
20. Guo, X.; Wu, Y.; Xie, L. Modified brain storm optimization algorithm for multimodal optimization. In Proceedings of the 5th International Conference on Advances in Swarm Intelligence, ICSI 2014, Hefei, China, 17–20 October 2014; pp. 340–351.
21. Pourpanah, F.; Wang, R.; Wang, X.; Shi, Y.; Yazdani, D. mBSO: A Multi-Population Brain Storm Optimization for Multimodal Dynamic Optimization Problems. In Proceedings of the 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 6–9 December 2019; pp. 673–679.
22. Yue, C.; Qu, B.; Yu, K.; Liang, J.; Li, X. A novel scalable test problem suite for multimodal multiobjective optimization. *Swarm Evol. Comput.* **2019**, *48*, 62–71. [[CrossRef](#)]
23. Yue, C.; Qu, B.; Liang, J. A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems. *IEEE Trans. Evol. Comput.* **2018**, *22*, 805–817. [[CrossRef](#)]
24. Zitzler, E.; Thiele, L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.* **1999**, *3*, 257–271. [[CrossRef](#)]
25. Shi, Y. Brain storm optimization algorithm. In Proceedings of the Advances in Swarm Intelligence—Second International Conference, ICSI 2011, Chongqing, China, 12–15 June 2011; pp. 303–309.
26. Deb, K.; Tiwari, S. Omni-optimizer: A procedure for single and multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*; Coello, C.A.C., Aguirre, A.H., Zitzler, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3410, pp. 47–61.
27. Zhang, J.-X.; Chu, X.-K.; Yang, F.; Qu, J.-F.; Wang, S.-W. Multimodal and multi-objective optimization algorithm based on two-stage search framework. *Appl. Intell.* **2022**, *52*, 12470–12496. [[CrossRef](#)]
28. Li, G.; Zhou, T. A multi-objective particle swarm optimizer based on reference point for multimodal multi-objective optimization. *Eng. Appl. Artif. Intell.* **2022**, *107*, 104523. [[CrossRef](#)]
29. Li, W.; Yao, X.; Zhang, T.; Wang, R.; Wang, L. Hierarchy Ranking Method for Multimodal Multiobjective Optimization With Local Pareto Fronts. *IEEE Trans. Evol. Comput.* **2023**, *27*, 98–110. [[CrossRef](#)]
30. Hu, Z.B.; Zhou, T.; Su, Q.H.; Liu, M.F. A niching backtracking search algorithm with adaptive local search for multimodal multiobjective optimization. *Swarm Evol. Comput.* **2022**, *69*, 101031. [[CrossRef](#)]
31. Hu, Y.; Wang, J.; Liang, J.; Wang, Y.L.; Ashraf, U.; Yue, C.T.; Yu, K.J. A two-archive model based evolutionary algorithm for multimodal multi-objective optimization problems. *Appl. Soft Comput.* **2022**, *119*, 108606. [[CrossRef](#)]
32. Gao, W.; Xu, W.; Gong, M.; Yen, G.G. A decomposition-based evolutionary algorithm using an estimation strategy for multimodal multi-objective optimization. *Inf. Sci.* **2022**, *606*, 531–548. [[CrossRef](#)]
33. Han, H.; Liu, Y.; Hou, Y.; Qiao, J. Multi-modal multi-objective particle swarm optimization with self-adjusting strategy. *Inf. Sci.* **2023**, *629*, 580–598. [[CrossRef](#)]

34. Ming, F.; Gong, W.Y.; Wang, L.; Gao, L. Balancing Convergence and Diversity in Objective and Decision Spaces for Multimodal Multi-Objective Optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **2022**, *7*, 474–486. [[CrossRef](#)]
35. Dang, Q.-L.; Xu, W.; Yuan, Y.-F. A Dynamic Resource Allocation Strategy with Reinforcement Learning for Multimodal Multi-objective Optimization. *Mach. Intell. Res.* **2022**, *19*, 138–152. [[CrossRef](#)]
36. Ji, H.; Chen, S.; Fan, Q. Zoning Search and Transfer Learning-based Multimodal Multi-objective Evolutionary Algorithm. In Proceedings of the 2022 IEEE Congress on Evolutionary Computation, CEC 2022, Padua, Italy, 18–23 July 2022; Institute of Electrical and Electronics Engineers Inc.: Padua, Italy, 2022.
37. Zhou, T.; Hu, Z.B.; Zhou, Q.; Yuan, S.X. A novel grey prediction evolution algorithm for multimodal multiobjective optimization. *Eng. Appl. Artif. Intell.* **2021**, *100*, 104173. [[CrossRef](#)]
38. Yue, C.T.; Suganthan, P.N.; Liang, J.; Qu, B.Y.; Yu, K.J.; Zhu, Y.S.; Yan, L. Differential evolution using improved crowding distance for multimodal multiobjective optimization. *Swarm Evol. Comput.* **2021**, *62*, 100849. [[CrossRef](#)]
39. Wilcoxon, F. Individual Comparisons by Ranking Methods. *Biom. Bull.* **1945**, *1*, 80–83. [[CrossRef](#)]
40. Friedman, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.