Machine Intelligence Research
www.mi-research.net

19(2), April 2022, 138-152
DOI: 10.1007/s11633-022-1314-7

A Dynamic Resource Allocation Strategy with Reinforcement Learning for Multimodal Multi-objective Optimization

Qian-Long Dang Wei Xu Yang-Fei Yuan

School of Mathematics and Statistics, Xidian University, Xi'an 710126, China

Abstract: Many isolation approaches, such as zoning search, have been proposed to preserve the diversity in the decision space of multimodal multi-objective optimization (MMO). However, these approaches allocate the same computing resources for subspaces with different difficulties and evolution states. In order to solve this issue, this paper proposes a dynamic resource allocation strategy (DRAS) with reinforcement learning for multimodal multi-objective optimization problems (MMOPs). In DRAS, relative contribution and improvement are utilized to define the aptitude of subspaces, which can capture the potentials of subspaces accurately. Moreover, the reinforcement learning method is used to dynamically allocate computing resources for each subspace. In addition, the proposed DRAS is applied to zoning searches. Experimental results demonstrate that DRAS can effectively assist zoning search in finding more and better distributed equivalent Pareto optimal solutions in the decision space.

Keywords: Multimodal multi-objective optimization (MMO), dynamic resource allocating strategy (DRAS), reinforcement learning (RL), decision space partition, zoning search.

Citation: Q. L. Dang, W. Xu, Y. F. Yuan. A dynamic resource allocation strategy with reinforcement learning for multimodal multi-objective optimization. *Machine Intelligence Research*, vol.19, no.2, pp.138–152, 2022. http://doi.org/10.1007/s11633-022-1314-7

1 Introduction

In real-world applications, many multi-objective optimization problems (MOPs) such as path planning[1], network optimization[2], structure optimization of neural networks[3], and multi-objective knapsack optimization[4] have multimodal properties, i.e., multiple Pareto optimal sets (PSs) that correspond to the same Pareto front (PF). Deb[5] defined these kinds of problems as multimodal multi-objective problems (MMOPs). Traditional multi-objective evolutionary algorithms (MOEAs)[6–8] usually improve the PF distribution in the objective space, which often leads to poor diversity of Pareto sets. Therefore, these MOEAs cannot solve the MMOPs.

In order to alleviate the above problem, two kinds of methods are proposed to keep the diversity of the decision space and the objective space. The first method is soft isolation, such as fitness sharing[9, 10], crowding[11, 12] and speciation[13]. Although these methods can effectively maintain diversity, the degradation of population diversity is inevitable due to increased environmental selection pressure. To solve this issue, hard isolation, i.e., zoning search strategy[14], is proposed as the second method. The total decision space is divided into several subspaces, which can search more widely in each subspace.

Although the hard isolation approach can effectively improve the diversity of decision space, there are two problems. Firstly, each subspace allocates the same computing resources without considering the difficulty of each subspace. If there are a few Pareto optimal solutions in some subspaces, a lot of computing resources will be wasted. Secondly, computing resources are not allocated reasonably according to the evolution state of each subspace. Thus, how to define the potential based on the difficulty and evolution state of each subspace is a very meaningful research problem. In addition, allocating the computing resources of each subspace adaptively according to the feedback information is also an urgent problem. The specific shortcomings are as follows:

1) The recent zoning search strategy allocates the same computing resources to all subspaces, which fails to consider the difficulties of subspaces. An illustrative example is presented in Fig. 1, where there is no PS curve in the two subspaces while the same computing resources are allocated to the two subspaces. The above facts inspired us to allocate computing resources based on the

Research Article
Manuscript received April 15, 2021; accepted August 19, 2021; published online January 7, 2022
Recommended by Associate Editor Long Cheng
Colored figures are available in the online version at https://link.springer.com/journal/11633

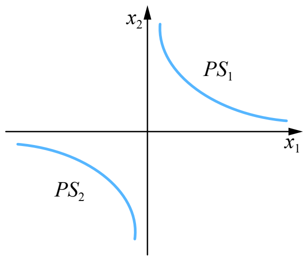


Fig. 1 Subspaces with different difficulties. The first and third quadrants have the same number of PSs, and the second and fourth quadrants have no PS.

difficulties of subspaces to achieve better results.

2) For the MMOP, the evolution states of subspaces with the same computational difficulty may be different. Specifically, Fig. 2 is an example that illustrates the different evolution states. The subspaces corresponding to the PS_1 curve obtain poor results, while the subspaces corresponding to the PS_2 curve get excellent results. In order to tackle this limitation, computing resources are dynamically allocated based on the evolution states. In this way, the limited resources can be fully utilized.

3) According to the analysis of the above two shortcomings, the difficulties and evolution states of subspaces are used as feedback information to allocate computing resources. However, the allocation of computing resources reasonably according to the feedback information is the key problem. In this paper, the computing resource allocation process is controlled by the reinforcement learning method.

In order to solve the above problems, a dynamic resource allocation strategy with reinforcement learning is proposed, called DRAS, which combines subspace's relative contribution with improvement to calculate the potentials of subspaces. Moreover, the computing resources of each subspace are dynamically allocated by the reinforcement learning method. In addition, the proposed DRAS is applied to zoning searches. The main contributions are listed as follows:

1) By introducing the concept of hypervolume, the subspaces′ relative contribution and improvement are obtained in the evolution process to measure the aptitudes

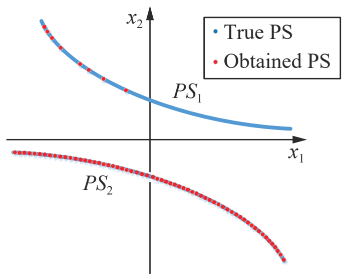


Fig. 2 Subspaces with different evolution states. A large number of PSs in the third and fourth quadrants have been found, and only some PSs in the first and second quadrants have been found.

of subspaces, which can reflect the difficulties and potentials of subspaces.

2) The reinforcement learning method is used to control the computing resource allocation process by the potentials of subspaces. In this way, the limited computing resources can be used efficiently.

3) The proposed DRAS is applied to zoning search[14], named DRAS-ZS_Ring_PSO_SCD. Compared with six multimodal multi-objective evolutionary algorithms (MMOEAs), the experimental results demonstrate that DRAS-ZS_Ring_PSO_SCD is efficient and competitive.

The rest of this paper is arranged as follows. Section 2 reviews the related work on the definition of MOP, the definition of MMOP, resource allocation in evolutionary algorithms, and reinforcement learning. Section 3 presents the details of DRAS. The experimental results are presented in Section 4. Finally, Section 5 concludes this paper.

2 Related work

2.1 Definition of MOP

Without loss of generality, an MOP[15] can be expressed as follows:

$$\min_{\boldsymbol{x}\in\boldsymbol{S}} f(\boldsymbol{x}) =(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))^{\rm T},$$
$$x_j \in (x_j^{\rm low}, x_j^{\rm high}),\ j = 1, 2, \cdots, n \qquad (1)$$

where $\boldsymbol{S}$ ($\boldsymbol{S} \subset \boldsymbol{R}^n$) represents the decision space; $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^{\rm T}$ denotes an n-dimensional vector; $f_i(\boldsymbol{x})$, $i = 1, 2, \cdots, m$, is the i-th objective to be minimized, and m is the number of objective functions; $x_j^{\rm low}$ and $x_j^{\rm high}$ are the lower and upper bounds of x_j, respectively; n is the dimension of the MOP.

2.2 Definition of MMOP

For the MMOP[5], the Pareto set comprises many disjoint subsets in the decision space. As shown in Fig. 3, the two curves in the (x_1, x_2) decision space represent two Pareto subsets, PS_1 and PS_2, which correspond to the same PF in the (f_1, f_2) objective space. The mapping from the two red circles in decision space to the red circle in objective space shows that there are solutions in PS_1 and PS_2. Although the two red circles belong to disjoint Pareto subsets, they map to the same objective function. Therefore, a new challenge is to obtain different PSs which correspond to the same PF for MMOPs.

In order to solve this challenge, many MMOEAs are proposed. According to the framework, these algorithms can be divided into the following three categories:

1) Framework based on nondominated sorting: Given that the diversity of the population is maintained, DN-NSGAII proposed by Liang et al.[16] can obtain more PSs. Maree et al.[17] combined a multi-objective valley cluster-

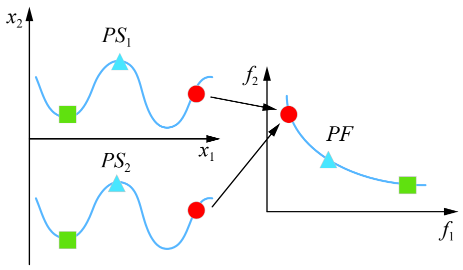


Fig. 3 Illustration of multimodel multi-objective problem. Two PSs correspond to one PF.

ing algorithm with the multi-objective algorithm MAM-aLGaM to solve MMOPs. Yue et al.[18] adopted a ring topology to obtain better diversity for MMOPs.

2) Framework based on decomposition: Tanabe and Ishibuchi[19, 20] proposed a decomposition-based MMOEA and a new framework to solve MMOPs. Peng and Ishibuchi[21] designed an efficient MMOEA based on a clearing mechanism and a greedy removal strategy.

3) Framework based on indicator: Tanabe and Ishibuchi[22] proposed a niching indicator-based MMOEA to solve the MMOP with more than three objectives. This method calculates the fitness of individuals and their nearest individuals to maintain diversity.

2.3 Resource allocation in evolutionary algorithms

Generally, when the difficulty of multiple subproblems or subtasks is inconsistent, they will face the problem of computing resource allocation, such as multi-objective optimization, co-evolution optimization, and multitask optimization. Over the past decades, the resource allocation strategies are mainly designed for multi-objective optimization[23–32], large-scale cooperative co-evolutionary optimization[33–40], and multitask optimization[41]. According to the conclusion in [25], the above resource allocation strategies include online and offline strategies.

1) Online resource allocation (ONRA): The computing resources are allocated dynamically according to the online feedback information.

2) Offline resource allocation (OFRA): The computing resources are allocated according to the pre-calculated task difficulty.

In MOPs, most of the research work[23–30], focuses on the improvement of subproblems, which only use the subproblem-level information instead of the overall population information. Although the work in [31] emphasizes the contribution of subproblems, it ignores the improvement of subproblems. With the aim to solve this issue, a mixed approach in [32] is proposed according to contribution rates and improvement rates of subproblems simultaneously. In cooperative coevolution, the resource allocation method in [38–40] allocates computing resources according to the improvement of subproblems. In addition, an improved resource allocation method in [37] is proposed according to accumulating contributions. In multitask optimization, an online DRAS[41] is designed based on the improvement of subtasks.

2.4 Reinforcement learning

Reinforcement learning (RL) is a field of machine learning, which emphasizes how to act based on the environment to maximize the expected benefits. As illustrated in Fig. 4, the RL algorithm obtains the updating state and reward from the environment. Afterward, the agent takes action on the environment. Over the last decades, RL algorithm has been applied in diverse fields, such as computer vision[42], financial investment[43], automatic driving[44], natural language processing[45], robot control[46], and building energy[47].

The problem of RL is normalized into the form of the Markov decision process (MDP) by more rigorous mathematical language. MDP is described by a tuple $(\mathcal{S},\mathcal{A},p,R,\gamma)$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, p is the state transition probability, R is the reward function, and γ is the discount factor. The goal of RL is to obtain the optimal strategy in the process of interaction between agent and environment. A policy is a mapping from state to action and is defined as follows:

$$\pi(a|s)=p[\mathcal{A}_t=a|\mathcal{S}_t=s] \tag{2}$$

where a is an action in $\mathcal{A}_t$ and s is a state in $\mathcal{S}_t$. When given a policy π, the expected discounted return G_t is calculated as

$$G_t=R_{t+1}+\gamma\times R_{t+2}+\cdots=\sum_{k=0}^{\infty}\gamma^k\times R_{t+k+1} \tag{3}$$

where $\gamma\in[0,1]$ and R_{t+i} is the reward function of the $(t+i)$-th generation. When the expected discounted return G_t is obtained, the state-value function $v_\pi(s)$ and the action-value function $q_\pi(s,a)$ are updated as follows:

$$v_\pi(s)=\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k\times R_{t+k+1}|\mathcal{S}_t=s\right] \tag{4}$$

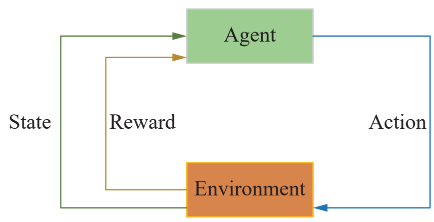


Fig. 4 Reinforcement learning model. The agent and environment interact repeatedly, and the optimal strategy is obtained.

$$q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k \times R_{t+k+1}|\mathcal{S}_t = s, \mathcal{A}_t = a\right] \tag{5}$$

where $\mathbb{E}_\pi$ is the expectation under the policy π.

3 Proposed approach

This section proposes DRAS in detail. Moreover, DRAS is applied to zoning searches.

3.1 Subspace′s relative contribution and improvement

This subsection defines the difficulties and evolution states of subspaces through the relative contribution and improvement, and then solves the two flaws of the zoning search strategy. In addition, the concept of hypervolume[48] is introduced to compare the solution of multi-objective optimization. As shown in Fig. 5, this is an example to illustrate the concept of hypervolume. The red dot represents the reference point. The yellow, blue, and green matrices represent the hypervolume of points A, B, and C, respectively. Specifically, the larger the hypervolume of the solution, the better its quality.

1) Subspace′s relative contribution: The relative contribution of the i-th subspace is defined as the proportion of the sum of the hypervolume of all solutions in the subspace to the sum of the hypervolume of solutions in the total decision space:

$$\Delta_{contri}^{i,t} = \frac{\sum\limits_{\boldsymbol{x}^t \in \boldsymbol{S}_i} HV(\boldsymbol{x}^t)}{\sum\limits_{\boldsymbol{x}^t \in \boldsymbol{S}} HV(\boldsymbol{x}^t)} \tag{6}$$

where t represents the current generation; $HV(\boldsymbol{x}^t)$ denotes the hypervolume of solution $\boldsymbol{x}^t$ in the objective space; $\boldsymbol{S}_i$ represents the i-th subspace. Generally, a small $\Delta_{contri}^{i,t}$ means that there is a small number of true PS in the i-th subspace. Therefore, it is hopeful to be assigned with less computing resources to this subspace for avoiding waste. In contrast, a large $\Delta_{contri}^{i,t}$ indicates that the i-th subspace has many true PS, requiring more computing resources. Consequently, limited computing resources can be used efficiently.

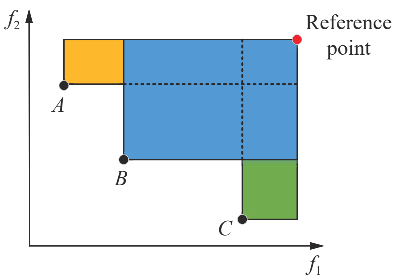


Fig. 5 Example for understanding the hypervolume. The hypervolume of points A, B, and C are shown in the rectangular area.

2) Subspace′s relative improvement: At the generation t, the relative improvement induced by the i-th subspace is defined as

$$\Delta_{improv}^{i,t} = \sum_{\boldsymbol{x}^t \in \boldsymbol{S}_i} \Delta_{improv}(\boldsymbol{x}^t) \tag{7}$$

where $\Delta_{improv}(\boldsymbol{x}^t)$ represents the improvement of $\boldsymbol{x}^t$ and can be defined as follows:

$$\Delta_{improv}(\boldsymbol{x}^t) = \frac{HV(\boldsymbol{x}^{t-1}) - HV(\boldsymbol{x}^t)}{HV(\boldsymbol{x}^{t-1})}. \tag{8}$$

The purpose of (7) is to accurately quantify the improvement degree of the i-th subspace. If the improvement of $\Delta_{improv}^{i,t}$ is large, the i-th subspace has great development potential, and vice versa. In this way, the potential of the total decision space can be exploited to the greatest extent.

3.2 Dynamic resource allocation strategy with reinforcement learning

Inspired by the parameter control strategy in [49], a dynamic resource allocation strategy based on RL is proposed. The MMOEA is regarded as the environment, and the subspace is the agent. The subspace′s relative contribution and improvement are the feedback information from the environment to the agent. The amount of computing resource allocation in subspace is the agent′s action to the environment. Thus, the dynamic resource allocation process is the control process of the RL method. In addition, at the $(g+1)$-th generation, the fitness contribution rate and the improvement rate are calculated by the subspace′s relative contribution and improvement over the last L generations, respectively.

$$FCR_{i,g+1} = \frac{\sum_{t=g-L+1}^{g}\gamma^{(g+1-t)}(\Delta_{contri}^{i,t}+\mu)}{L} \tag{9}$$

$$FIR_{i,g+1} = \frac{\sum_{t=g-L+1}^{g}\gamma^{(g+1-t)}(\Delta_{improv}^{i,t}+\mu)}{\sum_{i=1}^{N}\sum_{t=g-L+1}^{g}\gamma^{(g+1-t)}(\Delta_{improv}^{i,t}+\mu)} \tag{10}$$

where N is the number of subspaces; $\gamma \in [0,1]$ is a discount factor; μ is a small positive number to avoid the denominator to be zero and is set to $\mu = 1.0 \times 10^{-20}$; $\Delta_{contri}^{i,t}$ and $\Delta_{improv}^{i,t}$ are defined in (6) and (7). In (9), the fitness contribution rate can be reflected by the accumulated feedbacks induced by $\Delta_{contri}^{i,t}$ over the last L generations. Similarly, the fitness improvement rate is measured by the accumulated feedbacks induced by $\Delta_{improv}^{i,t}$ during the last L generations in (10).

By combining the fitness contribution rate and the fitness improvement rate, the aptitude of each subspace is defined as follows:

$$APT_{i,g+1} = \frac{FCR_{i,g+1} + FIR_{i,g+1}}{2}. \tag{11}$$

For different test functions and evolution phases, the difficulties and the potentials of the subspaces are different. For simplicity, the difficulties and the potentials of the subspaces are considered equally important. In (11), the greater contribution and improvement of a subspace, the more computing resources are allocated.

3.3 Applying DRAS to zoning search

Algorithm 1. DRAS-ZS_Ring_PSO_SCD
Input: Multimodal multi-objective optimization problem $MMOP$, number of subspaces N, maximum number of function evaluations $maxFEs$, subspace population size λ.
Output: Offspring population OFP.
1) The decision space S is divided into N subspaces, i.e., $S = \{S_1, S_2, \cdots, S_N\}$;
2) Generate an initial parent population $PAP = \{PAP_1, \cdots, PAP_N\}$, $PAP_i = \{x_1, x_2, \cdots, x_N\}$;
3) Set the aptitude $APT = \{\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N}\}$;
4) FEs $=FE$s $+ \lambda \times N$;
5) **While** $FEs \leq maxFEs$ **do**
6) The zoning search adopts PAP and APT to generate off-spring population OFP and computing resources used C_r;
7) $FEs = FEs + C_r$;
8) Calculate $\Delta_{contri}^{i,t}$ and $\Delta_{improv}^{i,t}$ according to (6) and (7);
9) Obtain $APT = \{APT_{1,t}, APT_{2,t}, \cdots, APT_{N,t}\}$ according to (9)−(11);
10) **End While**

The proposed DRAS is applied to the zoning search[14], named DRAS-ZS_Ring_PSO_SCD, which is summarized in Algorithm 1. Firstly, the decision space S is divided into N subspaces based on the zoning search approach. Secondly, an initial parent population PAP and the aptitude APT are generated. Thirdly, the computing resources of each subspace are dynamically allocated based on the feedback information in Steps 5)−10). More specifically, the zoning search algorithm adopts the allocated APT to generate the offspring population OFP in Step 6). Finally, according to the subspace's relative contribution and improvement, the aptitude of each subspace is calculated in each generation to guide the allocation of computing resources in Steps 8)−9).

4 Experimental studies

4.1 Test problems and performance indicators

The 22 test problems on MMO are designed by Liang et al.[50] The parameters settings of the test problems are summarized in Table 1. Specifically, x^l and x^u are the lower and upper bounds of the decision variables, respectively. F_{rep} is the reference point for hypervolume calculation, and PSs is the number of PS.

In the MOEAs, some performance metrics[51, 52] have been proposed. However, MMO pays more attention to PSs fitting in the decision space, so these performance indicators are not suitable for evaluating the performance of MMOEAs. Therefore, the hypervolume (HV)[51] and the Pareto set proximity (PSP)[18] are used to perform the assessment of MMOEAs. Specifically, HV is used to evaluate the obtained PF in the objective space, and PSP is utilized to evaluate the obtained PSs in the decision space. Because MMOP is to find multiple PSs in the de-

Table 1 Parameters settings of the test problems

Problems	x^l	x^u	F_{rep}	PSs
MMF1	[1 −1]	[3 1]	[1.1 1.1]	2
MMF1_z	[1 −1]	[3 1]	[1.1 1.1]	2
MMF1_e	[1 −20]	[3 20]	[1.1 1.1]	2
MMF2	[0 0]	[1 2]	[1.1 1.1]	2
MMF3	[0 0]	[1 1.5]	[1.1 1.1]	2
MMF4	[−1 0]	[1 2]	[1.1 1.1]	4
MMF5	[1 −1]	[3 3]	[1.1 1.1]	4
MMF6	[1 −1]	[3 2]	[1.1 1.1]	4
MMF7	[1 −1]	[3 1]	[1.1 1.1]	2
MMF8	[−π 0]	[π 9]	[1.1 1.1]	4
MMF9	[0.1 0.1]	[1.1 1.1]	[1.21 11]	2
MMF10	[0.1 0.1]	[1.1 1.1]	[1.21 13.2]	1
MMF11	[0.1 0.1]	[1.1 1.1]	[1.21 15.4]	1
MMF12	[0 0]	[1 1]	[1.54 1.1]	4
MMF13	[0.1 0.1 0.1]	[1.1 1.1 1.1]	[1.54 15.4]	1
MMF14	[0 0 0]	[1 1 1]	[2.2 2.2 2.2]	2
MMF14_a	[0 0 0]	[1 1 1]	[2.2 2.2 2.2]	2
MMF15	[0 0 0]	[1 1 1]	[2.5 2.5 2.5]	1
MMF15_a	[0 0 0]	[1 1 1]	[2.5 2.5 2.5]	1
SYM PART simple	[−20 −20]	[20 20]	[4.4 4.4]	9
SYM PART rotated	[−20 −20]	[20 20]	[4.4 4.4]	9
Omni-test	[0 0 0]	[6 6 6]	[4.4 4.4]	27

cision space, PSP is the main performance metric, and HV is the auxiliary performance metric.

4.2 Experimental settings

In this paper, seven competitive algorithms which are MOEA/D[6], NSGAII[7], ZS-MO_Ring_PSO_SCD[14], DN-NSGAII[16], MO_Ring_PSO_SCD[18], SSMOPSO[53], and DRAS-ZS_Ring_PSO_SCD are selected as comparison algorithms.

For seven comparison algorithms, the population size is 800, and the maximum number of evaluations is 160 000. Each algorithm runs 25 times independently. According to the conclusion in [14], the number of subspaces N can be set to 2, 4, and 6. For two-dimensional and three-dimensional test functions, N is set to 4 and 6 respectively, and the algorithm has the best performance. Moreover, to make a more accurate analysis, Friedman's test[54] is utilized to identify the performance of each algorithm. The signs "+", "−", and "≈" denote that the performance of the proposed algorithm is better, worse, and not much different than the comparison algorithm. In addition, all comparison algorithms are programmed in Matlab (R2017b).

4.3 Comparison with other algorithms

In the experiment, seven comparison algorithms are tested on 22 test problems in terms of HV and PSP. Moreover, the test problems are divided into two groups based on the difficulties of the subspaces. In the first test problem group, the difficulty of each subspace is different, while the difficulty is the same in the second test problem group.

Tables 2 and 3 show the mean and standard deviation of the seven comparison algorithms on two groups of test problems to HV. As shown in Table 2, DRAS-ZS_Ring_PSO_SCD surpasses the six other comparison

Table 2 HV values of seven algorithms on the first test problem group

Problems	DRAS-ZS_Ring_PSO_SCD	MOEA/D		NSGAII		ZS-MO_Ring_PSO_SCD		DN-NSGAII		MO_Ring_PSO_SCD		SSMOPSO	
MMF1_z	**8.81E−01** (**3.74E−05**)	2.18E−01 (5.63E−04)	+	8.73E−01 (5.63E−05)	+	8.76E−01 (4.72E−05)	+	8.72E−01 (6.36E−05)	+	8.75E−01 (4.59E−05)	+	8.75E−01 (2.06E−04)	+
MMF1_e	8.74E−01 (8.63E−04)	7.99E−01 (5.74E−05)	+	8.72E−01 (3.67E−04)	+	8.72E−01 (7.23E−04)	+	8.72E−01 (5.87E−04)	+	8.73E−01 (4.82E−04)	+	**8.75E−01** (**3.21E−04**)	−
MMF3	8.73E−01 (8.26E−04)	**8.76E−01** (**2.99E−05**)	−	8.72E−01 (8.53E−04)	+	8.72E−01 (8.64E−04)	+	8.72E−01 (9.23E−04)	+	8.73E−01 (6.08E−04)	≈	8.74E−01 (2.57E−04)	−
MMF7	8.75E−01 (3.68E−05)	8.76E−01 (2.12E−05)	−	**8.77E−01** (**1.79E−06**)	−	8.76E−01 (1.79E−05)	−	8.76E−01 (1.38E−06)	−	8.75E−01 (6.82E−05)	≈	8.74E−01 (9.03E−05)	+
MMF10	1.28E+01 (7.42E−03)	1.23E+01 (8.87E−03)	+	**1.30E+01** (**7.63E−03**)	−	1.28E+01 (7.10E−03)	≈	1.29E+01 (7.86E−03)	−	1.28E+01 (2.20E−03)	≈	1.29E+01 (4.70E−03)	−
MMF11	1.45E+01 (3.78E−05)	1.45E+01 (4.34E−05)	≈	**1.46E+01** (**5.27E−05**)	−	1.45E+01 (2.72E−04)	≈	1.45E+01 (5.16E−05)	≈	1.45E+01 (7.47E−04)	≈	1.45E+01 (2.90E−03)	≈
MMF12	1.57E+00 (1.74E−04)	1.57E+00 (3.18E−06)	≈	**1.58E+00** (**4.27E−06**)	−	1.57E+00 (1.60E−04)	≈	1.57E+00 (4.59E−06)	≈	1.57E+00 (1.85E−04)	≈	1.57E+00 (1.39E−04)	≈
MMF13	1.85E+01 (7.83E−04)	1.84E+01 (1.23E−04)	+	**1.86E+01** (**1.79E−05**)	−	1.85E+01 (8.85E−04)	≈	1.85E+01 (1.41E−04)	≈	1.84E+01 (9.25E−04)	+	1.85E+01 (7.95E−04)	≈
MMF15	**4.51E+00** (**8.54E−02**)	3.69E+00 (8.56E−02)	+	4.49E+00 (8.66E−02)	+	4.48E+00 (7.07E−02)	+	4.37E+00 (8.67E−02)	+	4.35E+00 (7.69E−02)	+	4.29E+00 (9.54E−02)	+
MMF15_a	4.62E+00 (8.63E−02)	3.73E+00 (1.11E−01)	+	**4.83E+00** (**9.73E−02**)	−	4.47E+00 (8.13E−02)	+	4.47E+00 (8.74E−02)	+	4.33E+00 (6.00E−02)	+	4.29E+00 (7.65E−02)	+
+			6		4		5		5		5		4
≈			2		0		4		3		5		3
−			2		6		1		2		0		3

algorithms on 6, 4, 5, 5, 5, and 4 test problems. Moreover, DRAS-ZS_Ring_PSO_SCD surpasses the six other comparison algorithms on 7, 2, 8, 6, 11, and 9 test problems in Table 3. Furthermore, this paper conducts Friedman′s test[54] on seven comparison algorithms based on 22 test problems. Table 4 presents the test results, and it shows that NSGAII has the best ranking, sequentially followed by DRAS-ZS_Ring_PSO_SCD, ZS-MO_Ring_PSO_SCD, DN-NSGAII, MOEA/D, SSMOPSO, and MO_Ring_PSO_SCD. It can be seen that DRAS-ZS_Ring_PSO_SCD performs worse than NSGAII. This is because DRAS mainly improves the performance of decision space. The performance of objective space mainly depends on the multi-objective handling technology. Finally, although DRAS-ZS_Ring_PSO_SCD does not have the best ranking, its performance is significantly better than the other five comparison algorithms.

Besides the HV, the indicator PSP is used to evaluate seven comparison algorithms. As shown in Table 5, the first group test problems are used, which have different difficulties in each subspace. DRAS-ZS_Ring_PSO_SCD surpasses the six other comparison algorithms on 12, 12, 12, 12, 12, and 10 test problems. This is because the DRAS can dynamically allocate computing resources based on the contribution and improvement degree of the subspaces, which can effectively improve the overall performance. Moreover, the second group test problems are

Table 3 HV values of seven algorithms on the second test problem group

Problems	DRAS-ZS_Ring_PSO_SCD	MOEA/D		NSGAII		ZS-MO_Ring_PSO_SCD		DN-NSGAII		MO_Ring_PSO_SCD		SSMOPSO	
MMF1	8.76E−01 (1.58E−05)	8.76E−01 (1.71E−05)	≈	**8.77E−01 (1.79E−05)**	−	8.76E−01 (4.29E−05)	≈	8.76E−01 (4.19E−06)	≈	8.75E−01 (4.61E−05)	+	8.75E−01 (2.13E−04)	+
MMF2	8.73E−01 (6.25E−04)	**8.76E−01 (2.03E−05)**	−	8.73E−01 (1.84E−03)	≈	8.72E−01 (5.66E−04)	+	8.71E−01 (1.60E−03)	+	8.72E−01 (3.42E−04)	+	8.73E−01 (5.43E−04)	≈
MMF4	5.44E−01 (5.66E−06)	5.43E−01 (8.38E−07)	+	**5.48E−01 (8.27E−06)**	−	5.43E−01 (7.45E−05)	+	5.43E−01 (8.12E−06)	+	5.42E−01 (1.28E−04)	+	5.42E−01 (1.95E−04)	+
MMF5	8.76E−01 (7.26E−06)	8.76E−01 (1.20E−05)	≈	**8.77E−01 (7.67E−05)**	−	8.76E−01 (4.76E−05)	≈	8.76E−01 (1.59E−05)	≈	8.75E−01 (3.37E−05)	+	8.75E−01 (2.71E−04)	+
MMF6	8.77E−01 (6.26E−05)	8.76E−01 (1.81E−04)	+	**8.79E−01 (7.35E−05)**	−	8.76E−01 (4.01E−05)	+	8.76E−01 (2.95E−06)	+	8.75E−01 (4.30E−05)	+	8.75E−01 (1.74E−04)	+
MMF8	4.24E−01 (4.33E−05)	**4.25E−01 (3.92E−05)**	−	4.24E−01 (7.53E−05)	≈	4.24E−01 (8.42E−05)	≈	4.24E−01 (6.40E−05)	≈	4.23E−01 (3.58E−04)	+	4.24E−01 (1.26E−04)	≈
MMF9	9.72E+00 (5.63E−05)	9.69E+00 (2.60E−05)	+	**9.74E+00 (7.43E−05)**	−	9.71E+00 (1.48E−04)	+	9.71E+00 (6.36E−05)	+	9.70E+00 (8.44E−04)	+	9.69E+00 (4.70E−03)	+
MMF14	3.14E+00 (5.28E−02)	2.11E+00 (8.68E−02)	+	**3.15E+00 (7.25E−02)**	−	3.15E+00 (9.54E−02)	−	3.14E+00 (9.67E−02)	≈	3.06E+00 (1.37E−01)	+	2.97E+00 (7.33E−02)	+
MMF14_a	3.19E+00 (1.68E−01)	2.79E+00 (1.70E−01)	+	**3.21E+00 (1.65E−01)**	−	3.17E+00 (1.29E−01)	+	3.20E+00 (1.67E−02)	−	3.00E+00 (1.35E−01)	+	3.10E+00 (4.14E−02)	+
SYM_PART simple	1.13E+01 (3.67E−05)	1.67E+01 (2.64E−05)	−	**1.69E+01 (7.25E−05)**	−	1.06E+01 (8.27E−04)	+	1.67E+01 (4.91E−05)	−	1.67E+01 (5.52E−04)	−	1.66E+00 (4.86E−04)	−
SYM_PART rotated	**1.68E+01 (7.26E−04)**	1.67E+01 (3.10E−05)	+	1.67E+01 (2.77E−05)	+	1.31E+00 (8.64E−04)	+	1.67E+01 (5.99E−05)	+	1.67E+01 (1.30E−03)	+	1.67E+01 (3.75E− 03)	+
Omni-test	**6.22E+01 (3.67E−03)**	5.28E+01 (1.20E−03)	+	5.46E+00 (3.76E−04)	+	6.20E+01 (7.70E−03)	+	5.28E+01 (5.23E−04)	+	5.28E+01 (6.00E−03)	+	5.28E+01 (9.64E−04)	+
+			7		2		8		6		11		9
≈			2		2		3		4		0		2
−			3		8		1		2		1		1

Table 4 Results obtained by the Friedman's test on 22 test problems on MMO to HV

Algorithms	Ranking
NSGAII	**2.295 5**
DRAS-ZS_Ring_PSO_SCD	3.000 0
ZS-MO_Ring_PSO_SCD	4.022 7
DN-NSGAII	4.045 5
MOEA/D	4.704 5
SSMOPSO	4.840 9
MO_Ring_PSO_SCD	5.090 9

utilized in Table 6. It is confirmed that DRAS-ZS_Ring_PSO_SCD can perform better than six other comparison algorithms on all test problems, except for the problems SYM-PART simple and Omni-test. Therefore, it can be concluded that the DRAS can allocate more computing resources to subspaces with greater performance improvement. Furthermore, this paper conducts Friedman's test[54] on seven comparison algorithms based on 22 test problems. Finally, Table 7 presents the test results showing that DRAS-ZS_Ring_PSO_SCD has the best ranking, which indicates that the DRAS is an effective method for MMOPs.

4.4 Visual comparison

To further verify the effectiveness of DRAS, two problems (i.e., MMF3 and Omni-test) are utilized. As shown in Fig. 6, these are the obtained PSs and the true PSs on problem MMF3. Clearly, it can be seen that seven algorithms can find two PS regions. In particular, DRAS-ZS_Ring_PSO_SCD can obtain better solutions than six other comparison algorithms in each PS. Therefore, DRAS-ZS_Ring_PSO_SCD has the best performance. For

Table 5 PSP values of seven algorithms on the first test problem group

Problems	DRAS-ZS_Ring_PSO_SCD	MOEA/D		NSGAII		ZS-MO_Ring_PSO_SCD		DN-NSGAII		MO_Ring_PSO_SCD		SSMOPSO	
MMF1_z	**1.71E+02**	3.00E+01	+	8.31E+01	+	1.64E+02	+	1.13E+02	+	9.86E+01	+	9.47E+01	+
	(5.48E+00)	(5.23E−01)		(4.30E+00)		(6.82E+00)		(5.92E+00)		(5.97E+00)		(5.87E+00)	
MMF1_e	5.97E+00	3.78E+00	+	3.93E+00	+	5.92E+00	+	5.74E+00	+	5.95E+00	+	**1.02E+01**	−
	(1.40E+00)	(1.44E−01)		(6.93E−01)		(9.60E−01)		(8.36E−01)		(1.40E+00)		**(6.01E−01)**	
MMF3	2.15E+02	8.86E+00	+	1.29E+02	+	2.11E+02	+	1.89E+02	+	2.20E+02	+	**2.84E+02**	−
	(2.53E+01)	(4.54E+00)		(4.63E+01)		(2.04E+01)		(6.21E+01)		(1.56E+01)		**(1.36E+01)**	
MMF7	**2.08E+02**	5.31E+00	+	7.42E+01	+	2.02E+02	+	1.18E+02	+	1.10E+02	+	5.78E+01	+
	(1.04E+01)	(2.26E+00)		(4.28E+00)		(8.38E+00)		(5.57E+00)		(6.24E+00)		(4.79E+00)	
MMF10	**6.22E+00**	3.48E−01	+	1.73E−03	+	6.09E+00	+	2.50E−03	+	6.17E+00	+	6.18E+00	+
	(2.32E−02)	(4.07E−01)		(1.05E−04)		(1.07E−01)		(1.24E−04)		(2.26E−02)		(2.53E−02)	
MMF11	**5.31E+00**	6.46E−02	+	2.36E+00	+	3.05E+00	+	3.03E+00	+	5.30E+00	+	4.09E+00	+
	(7.26E−02)	(2.75E−02)		(1.78E−02)		(2.12E−02)		(5.25E−02)		(8.73E−02)		(1.86E+00)	
MMF12	**4.51E+00**	5.65E−02	+	3.11E+00	+	4.34E+00	+	4.31E+00	+	4.42E+00	+	1.67E+00	+
	(2.18E+00)	(4.45E−02)		(3.10E+00)		(2.88E+00)		(3.89E+00)		(2.60E+00)		(2.16E+00)	
MMF13	**3.72E+00**	7.42E−01	+	7.28E−01	+	3.25E+00	+	1.27E+00	+	3.51E+00	+	2.69E+00	+
	(9.27E−01)	(4.00E−01)		(5.28E−01)		(7.73E−01)		(7.90E−03)		(6.73E−01)		(7.15E−01)	
MMF15	**6.67E+00**	9.96E+00	+	4.19E+00	+	6.63E+00	+	5.84E+00	+	6.57E+00	+	5.62E+00	+
	(8.26E−01)	(6.17E−01)		(5.72E−01)		(7.20E−01)		(7.36E−01)		(6.63E−01)		(3.11E−01)	
MMF15_a	**6.45E+00**	9.76E+00	+	5.20E+00	+	6.30E+00	+	6.30E+00	+	6.42E+00	+	6.31E+00	+
	(4.73E−01)	(5.13E−02)		(5.72E−01)		(4.57E−01)		(4.74E−01)		(3.96E−01)		(1.51E−01)	
+			**12**		**12**		**12**		**12**		**12**		**10**
≈			**0**		**0**		**0**		**0**		**0**		**0**
−			**0**		**0**		**0**		**0**		**0**		**2**

Table 6 PSP values of seven algorithms on the second test problem group

Problems	DRAS-ZS_Ring_PSO_SCD	MOEA/D		NSGAII		ZS-MO_Ring_PSO_SCD		DN-NSGAII		MO_Ring_PSO_SCD		SSMOPSO	
MMF1	**1.37E+02** **(6.19E+00)**	7.29E+00 (6.47E+00)	+	4.82E+01 (3.82E+00)	+	1.24E+02 (5.52E+00)	+	6.82E+01 (1.61E+00)	+	7.21E+01 (3.28E+00)	+	7.35E+01 (7.18E+00)	+
MMF2	**2.32E+02** **(1.84E+01)**	4.13E+00 (3.48E+00)	+	8.29E+01 (2.48E+01)	+	2.13E+02 (2.15E+01)	+	1.47E+02 (4.44E+01)	+	1.80E+02 (1.63E+01)	+	2.61E+02 (1.10E+01)	+
MMF4	**2.57E+02** **(4.03E+00)**	1.55E+00 (6.47E−01)	+	5.74E+01 (3.86E+00)	+	2.20E+02 (5.97E+00)	+	1.04E+02 (4.54E+00)	+	1.21E+02 (2.05E+00)	+	1.15E+02 (3.22E+00)	+
MMF5	**5.90E+01** **(4.58E+00)**	3.48E+00 (2.31E+00)	+	1.29E+01 (1.39E+00)	+	5.85E+01 (3.21E+00)	+	3.13E+01 (1.66E+00)	+	3.63E+01 (7.30E−01)	+	3.74E+01 (3.97E+00)	+
MMF6	**7.01E+01** **(3.30E+00)**	4.35E+00 (2.14E+00)	+	1.73E+01 (1.94E+00)	+	6.83E+01 (2.33E+00)	+	3.35E+01 (1.86E+00)	+	3.85E+01 (1.51E+00)	+	3.67E+01 (1.03E+00)	+
MMF8	**8.72E+01** **(1.79E+00)**	1.25E−01 (8.92E−02)	+	4.25E+01 (1.95E+00)	+	8.61E+01 (3.62E+00)	+	8.17E+00 (2.82E+00)	+	5.39E+01 (2.49E+00)	+	4.26E+01 (8.13E+00)	+
MMF9	**7.84E+01** **(1.59E+01)**	2.07E−01 (4.95E−02)	+	2.48E+02 (9.42E+00)	+	7.69E+02 (1.91E+01)	+	4.30E+02 (1.68E+01)	+	3.98E+02 (7.68E+00)	+	3.76E+02 (3.20E+01)	+
MMF14	**6.59E+01** **(3.79E−01)**	1.00E+01 (1.92E−01)	+	2.18E+01 (3.80E−01)	+	5.61E+01 (4.30E−01)	+	2.97E+01 (4.63E−01)	+	3.02E+01 (3.91E−01)	+	2.33E+01 (6.33E−01)	+
MMF14_a	**5.43E+01** **(8.11E−01)**	9.81E+00 (1.81E−01)	+	2.59E+01 (4.73E−01)	+	5.07E+01 (7.19E−01)	+	4.05E+01 (3.57E−01)	+	2.70E+01 (6.34E−01)	+	2.45E+01 (4.30E−01)	+
SYM_PART simple	5.70E+01 (1.03E+00)	4.69E−04 (3.79E−04)	+	2.47E+01 (1.06E+00)	+	3.38E+01 (1.89E+00)	+	**8.20E+01** **(1.73E+00)**	−	3.15E+01 (1.14E+00)	+	3.05E+01 (2.96E+00)	+
SYM_PART rotated	**6.39E+01** **(1.79E+00)**	1.63E−02 (1.80E−02)	+	2.74E+01 (1.06E+00)	+	3.05E+01 (1.75E+00)	+	5.98E+01 (9.04E−01)	+	2.66E+01 (1.29E+00)	+	2.58E+01 (1.62E+00)	+
Omni-test	1.99E+01 (3.18E+00)	3.35E−02 (3.40E−03)	+	1.60E+01 (2.83E+00)	+	1.28E+01 (3.97E+00)	+	**4.17E+01** **(4.22E+00)**	−	1.01E+01 (2.03E+00)	+	1.16E+01 (7.28E+00)	+
+			12		12		12		10		12		12
≈			0		0		0		0		0		0
−			0		0		0		2		0		0

the problem Omni-test, PSs obtained by seven algorithms are provided in Fig. 7. Specifically, the Omni-test problem is the most complex compared to the other 21 problems because it has 27 PSs. As shown in Fig. 7, DRAS-ZS_Ring_PSO_SCD, ZS-MO_Ring_PSO_SCD, DN-NSGAII, MO_Ring_PSO_SCD, and SSMOPSO can find 27, 27, 27, 26, and 27 PS regions, respectively. However, the DRAS-ZS_Ring_PSO_SCD can obtain better solutions than these algorithms in each PS. In addition, MOEA/D and NSGAII can only find one PS region.

Through the above visual comparisons, DRAS can al-

Table 7 Results obtained by the Friedman's test on 22 test problems on MMO to PSP

Algorithms	Ranking
DRAS-ZS_Ring_PSO_SCD	**1.590 9**
ZS-MO_Ring_PSO_SCD	2.840 9
MO_Ring_PSO_SCD	3.409 1
SSMOPSO	4.000 0
DN-NSGAII	4.113 6
NSGAII	5.727 3
MOEA/D	6.318 2

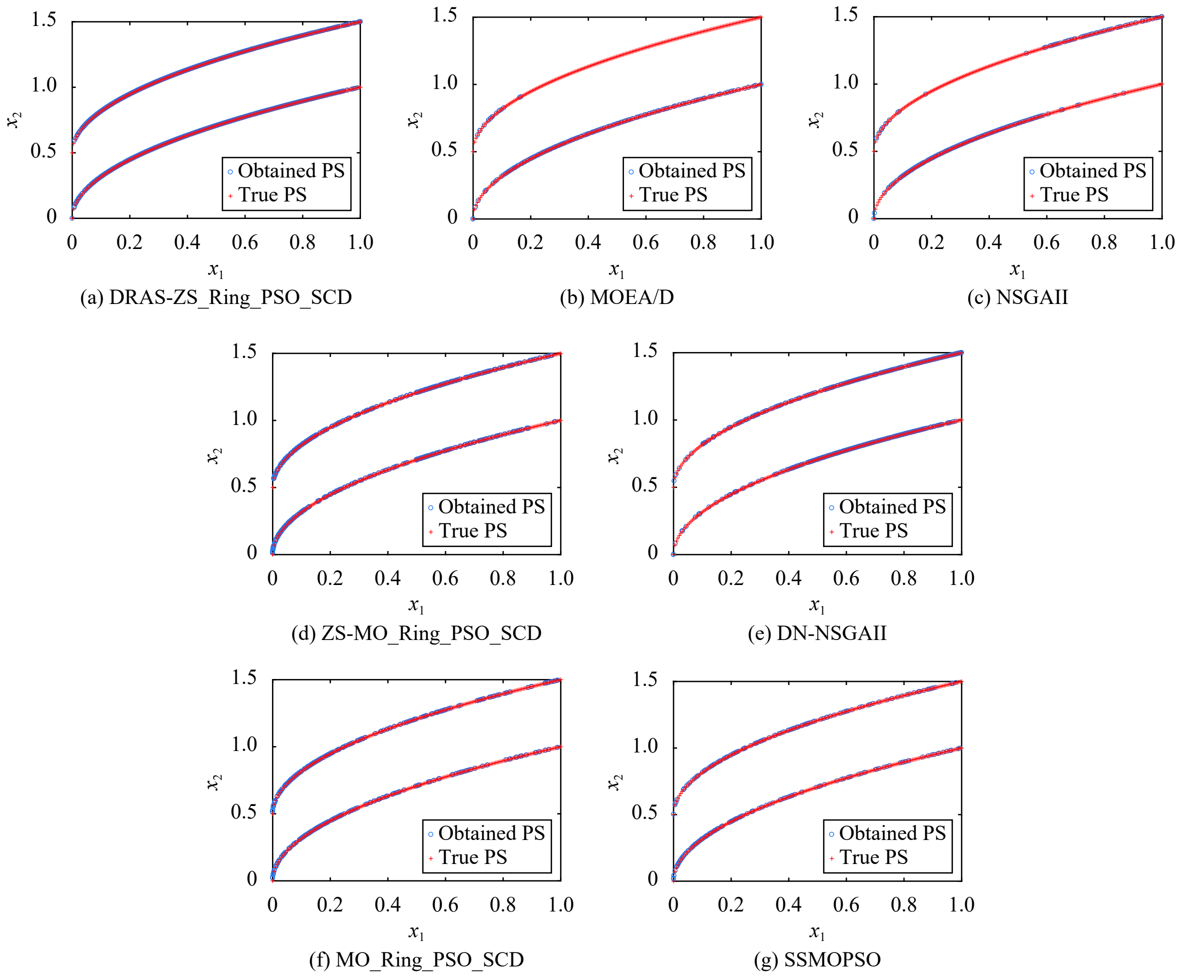


Fig. 6 Obtained PSs and the true PSs on problem MMF3 for seven comparison algorithms: (a) DRAS-ZS_Ring_PSO_SCD; (b) MOEA/D; (c) NSGAII; (d) ZS-MO_Ring_PSO_SCD; (e) DN-NSGAII; (f) MO_Ring_PSO_SCD; (g) SSMOPSO.

locate the computing resources to the improved and contributed regions so that MMOEAs can obtain more well-distributed PSs. Therefore, DRAS is an effective way to solve MMOPs.

4.5 Parameter analysis

DRAS introduces one parameter: the discount factor γ in (9) and (10). For the discount factor, a large γ can make full use of historical information. However, early historical information may have a negative impact on current resource allocation strategies. Moreover, a small γ cannot effectively utilize historical information. Therefore, we investigate how γ affects the performance of DRAS-ZS_Ring_PSO_SCD by testing five different γ: 0.1, 0.3, 0.5, 0.7, and 0.9.

As shown in Fig. 8, there are the results provided by DRAS-ZS_Ring_PSO_SCD with five different γ on MMF4, MMF11, MMF14, and SYM-PART rotated. When the discount factor is 0.7, DRAS-ZS_Ring_PSO_SCD has the best performance. The reason is as follows. A small γ cannot make full use of historical information, while a big γ uses too much historical information, which has a negative effect on the proposed DRAS.

4.6 Computational complexity

Fig. 9 presents the average runtime of five MMOEAs on 22 MMO test functions. As shown in Fig. 9, DN-NSGAII has the longest runtime, sequentially followed by SSMOPSO, DRAS-ZS_Ring_PSO_SCD, ZS-MO_Ring_PSO_SCD, and MO_Ring_PSO_SCD. Firstly, the runtime of DN-NSGAII and SSMOPSO is significantly higher than the other three MMOEAs. This is because DN-NSGAII uses a niching method to create the mating pool, which consumes much computation time. Moreover, SSMOPSO adopts the self-organizing map network to establish the neighborhood of the particles, and much computation time is used. Secondly, the zoning search strategy is applied to MO_Ring_PSO_SCD to get ZS-MO_Ring_PSO_SCD, and ZS-MO_Ring_PSO_SCD has longer runtime than MO_Ring_PSO_SCD. This is because the zoning search strategy causes an additional computing burden. Finally, DRAS is utilized to enhance the performance of ZS-MO_Ring_PSO_SCD. As shown in Fig. 9, the runtime of ZS-MO_Ring_PSO_SCD and

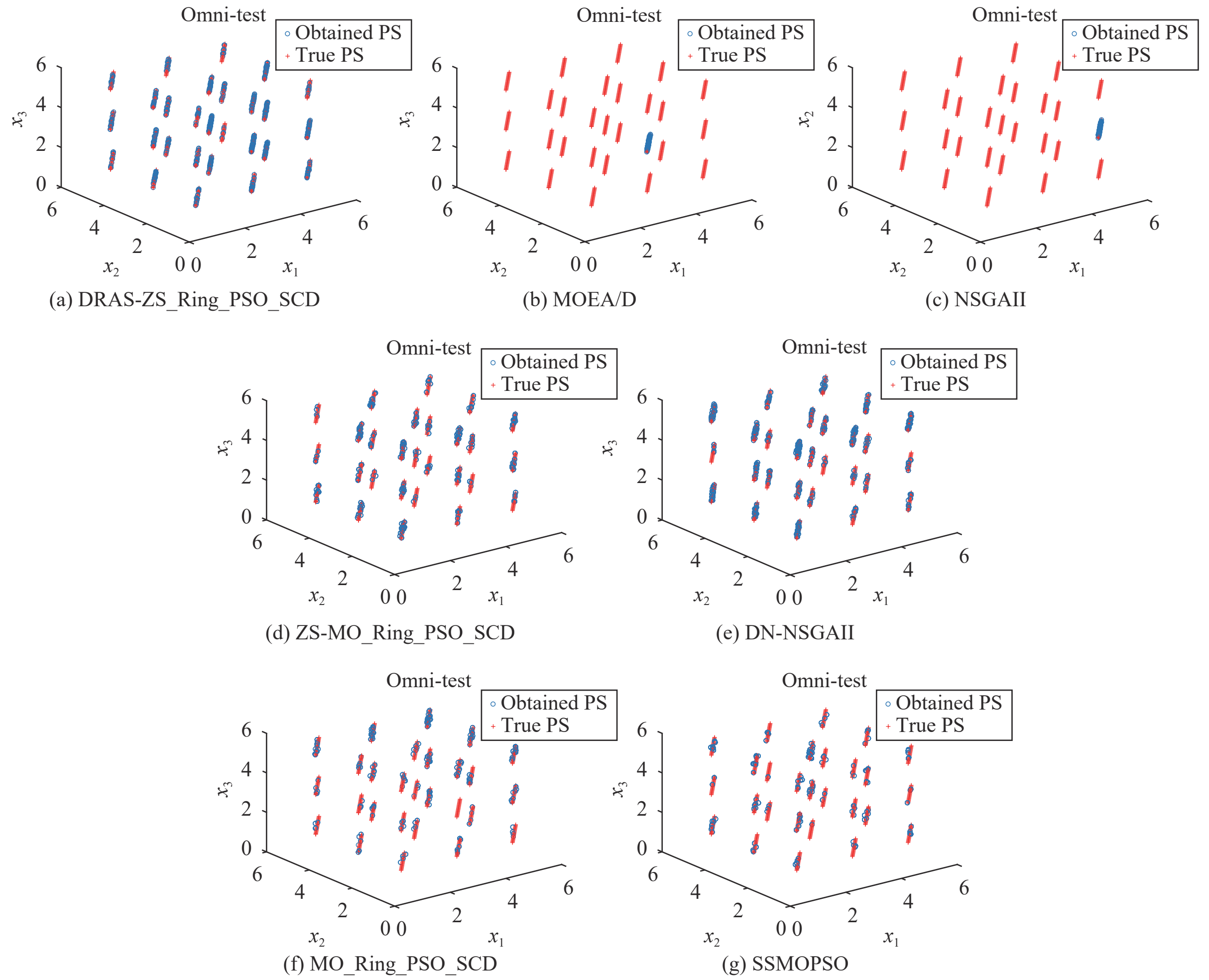


Fig. 7 Obtained PSs and the true PSs on problem Omni-test for seven comparison algorithms: (a) DRAS-ZS_Ring_PSO_SCD; (b) MOEA/D; (c) NSGAII; (d) ZS-MO_Ring_PSO_SCD; (e) DN-NSGAII; (f) MO_Ring_PSO_SCD; (g) SSMOPSO.

DRAS-ZS_Ring_PSO_SCD are almost similar, indicating that DRAS significantly improves the performance of ZS-MO_Ring_PSO_SCD without too much additional computing burden.

4.7 Imbalance problems

Twelve imbalance test problems on MMO are used as test problems. Seven MMOEAs, which are CPDEA[55], MO_Ring_PSO_SCD[18], DNEA[56], TriMOEA-TA&R[57], DN-NSGAII[16], Omni-optimizer[58], and DRAS-ZS_Ring_PSO_SCD, are selected as comparison algorithms. Each algorithm runs 40 times independently. For the two-objective, three-objective, and four-objective problems, the maximum number of evaluations is 18 000, 36 000, and 72 000, respectively, and the population size is set to 60, 120, and 240, respectively. In order to compare different algorithms on these test problems, we adopt inverted generational distance (IGDX)[59] as the performance metrics. IGDX can give a comprehensive quantification of both the convergence and the diversity in the decision space of the approximate solution set.

As shown in Table 8, the imbalance test problems are used. DRAS-ZS_Ring_PSO_SCD surpasses the six comparison algorithms on 9, 12, 12, 12, 12, and 12 test problems. This indicates that DRAS-ZS_Ring_PSO_SCD still achieves excellent performance on the imbalance test problems. Moreover, the difficulty of each PS is different in the unbalanced problems, and the proposed DRAS can efficiently allocate computing resources to improve the performance of the algorithm.

5 Conclusions

In this paper, a dynamic resource allocation strategy with reinforcement learning is proposed for MMO. The aptitudes of subspaces are defined by relative contribution and improvement. Moreover, computing resources are dynamically allocated by the RL method based on the aptitudes of the subspaces, which can make the subspace with larger potential allocate more computing resources. Finally, the experimental results indicate that DRAS is efficient and competitive.

In the future, there are still several issues that deserve to be investigated. Firstly, the potential of subspaces can be defined by new indicators. Secondly, since

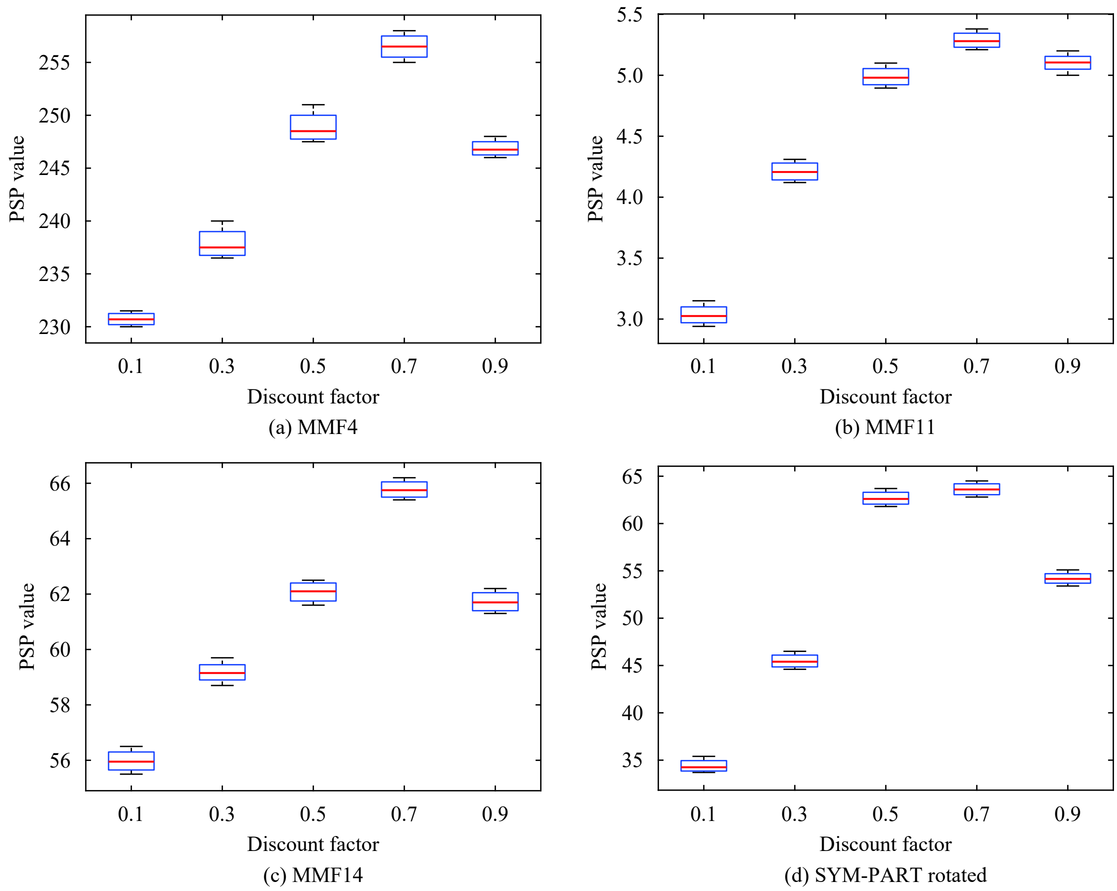


Fig. 8 Box plots of the PSP value derived from DRAS-ZS_Ring_PSO_SCD with different discount factors on MMF4, MMF11, MMF14, and SYM-PART rotated

Table 8 IGDX values of seven algorithms on imbalance test problems

Problems	DRAS-ZS_Ring_PSO_SCD	CPDEA		MO_Ring_PSO_SCD		DNEA		TriMOEA-TA&R		DN-NSGA-II		Omni-Optimizer	
IDMP-M2-T1	3.78E−03	**9.91E−04**	−	5.82E−02	+	3.20E−01	+	3.20E−01	+	1.70E−01	+	3.06E−01	+
IDMP-M2-T2	**9.75E−04**	1.04E−03	+	2.79E−03	+	2.53E−01	+	3.03E−01	+	1.86E−01	+	2.06E−01	+
IDMP-M2-T3	**1.27E−03**	1.53E−03	+	2.71E−03	+	8.80E−02	+	2.56E−01	+	2.22E−01	+	2.90E−01	+
IDMP-M2-T4	5.92E−03	**1.01E−03**	−	1.05E−01	+	3.20E−01	+	3.37E−01	+	3.37E−01	+	3.22E−01	+
IDMP-M3-T1	**3.25E−03**	5.64E−03	+	6.39E−02	+	4.08E−01	+	3.97E−01	+	3.08E−01	+	3.89E−01	+
IDMP-M3-T2	**4.97E−03**	5.66E−03	+	3.57E−02	+	3.60E−01	+	3.91E−01	+	3.55E−01	+	3.95E−01	+
IDMP-M3-T3	**6.01E−03**	6.17E−03	+	7.98E−03	+	2.67E−01	+	3.42E−01	+	2.74E−01	+	3.34E−01	+
IDMP-M3-T4	**3.16E−03**	5.62E−03	+	2.29E−02	+	4.05E−01	+	4.55E−01	+	4.61E−01	+	4.26E−01	+
IDMP-M4-T1	5.81E−03	**4.31E−03**	−	3.01E−01	+	5.93E−01	+	5.96E−01	+	4.99E−01	+	5.97E−01	+
IDMP-M4-T2	**3.84E−03**	4.35E−03	+	6.22E−02	+	5.20E−01	+	5.40E−01	+	5.55E−01	+	5.56E−01	+
IDMP-M4-T3	**4.41E−03**	4.78E−03	+	7.51E−03	+	4.84E−01	+	4.91E−01	+	3.42E−01	+	4.34E−01	+
IDMP-M4-T4	**4.25E−03**	4.38E−03	+	8.02E−03	+	5.10E−01	+	5.30E−01	+	5.28E−01	+	5.26E−01	+
+			9		12		12		12		12		12
≈			0		0		0		0		0		0
−			3		0		0		0		0		0

contribution and improvement to the subspace are treated equally, it is an interesting idea to dynamically control the ratio of contribution and improvement based on historical feedback. Finally, it would be very meaning-

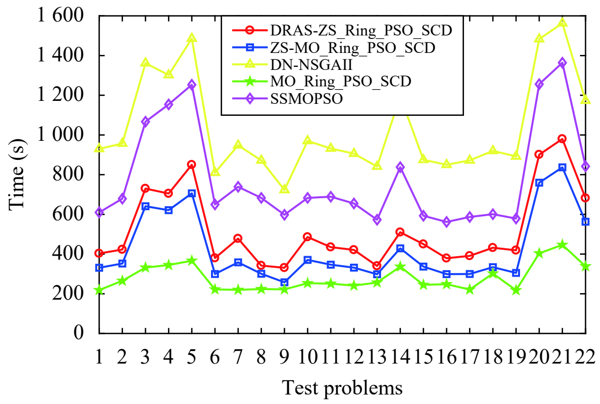


Fig. 9 Average runtime of five MMOEAs on 22 MMO test problems.

ful to apply the proposed method to the actual scene.

References

[1] S. L. Karri, L. C. De Silva, D. T. C. Lai, S. Y. Yong. Identification and classification of driving behaviour at signalized intersections using support vector machine. *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 480–491, 2021. DOI: 10.1007/s11633-021-1295-y.

[2] H. T. Ye, Z. Q. Li. PID neural network decoupling control based on hybrid particle swarm optimization and differential evolution. *International Journal of Automation and Computing*, vol. 17, no. 6, pp. 867–872, 2020. DOI: 10.1007/s11633-015-0917-7.

[3] W. Jia, W. Xia, Y. Zhao, H. Min, Y. X. Chen. 2D and 3D palmprint and palm vein recognition based on neural architecture search. *International Journal of Automation and Computing*, vol. 18, no. 3, pp. 377–409, 2021. DOI: 10.1007/s11633-021-1292-1.

[4] A. Jaszkiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002. DOI: 10.1109/TEVC.2002.802873.

[5] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999. DOI: 10.1162/evco.1999.7.3.205.

[6] Q. F. Zhang, H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007. DOI: 10.1109/TEVC.2007.892759.

[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002. DOI: 10.1109/4235.996017.

[8] S. Lu, Y. M. Li, B. X. Ding. Multi-objective dimensional optimization of a 3-DOF translational PKM considering transmission properties. *International Journal of Automation and Computing*, vol. 16, no. 6, pp. 748–760, 2019. DOI: 10.1007/s11633-019-1184-9.

[9] P. S. Oliveto, D. Sudholt, C. Zarges. On the benefits and risks of using fitness sharing for multimodal optimisation. *Theoretical Computer Science*, vol. 773, pp. 53–70, 2019. DOI: 10.1016/j.tcs.2018.07.007.

[10] C. Y. Lin, W. H. Wu. Niche identification techniques in multimodal genetic search with sharing scheme. *Advances in Engineering Software*, vol. 33, no. 11–12, pp. 779–791, 2002. DOI: 10.1016/S0965-9978(02)00045-5.

[11] R. Thomsen. Multimodal optimization using crowding-based differential evolution. In *Proceedings of Congress on Evolutionary Computation*, IEEE, Portland, USA, pp. 1382–1389, 2004. DOI: 10.1109/CEC.2004.1331058.

[12] M. Q. Li, D. Lin, J. S. Kou. A hybrid niching PSO enhanced with recombination-replacement crowding strategy for multimodal function optimization. *Applied Soft Computing*, vol. 12, no. 3, pp. 975–987, 2012. DOI: 10.1016/j.asoc.2011.11.032.

[13] J. P. Li, M. E. Balazs, G. T. Parks, P. J. Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary Computation*, vol. 10, no. 3, pp. 207–234, 2002. DOI: 10.1162/106365602760234081.

[14] Q. Q. Fan, X. F. Yan. Solving multimodal multiobjective problems through zoning search. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 8, pp. 4836–4847, 2021. DOI: 10.1109/TSMC.2019.2944338.

[15] K. Miettinen. *Nonlinear Multiobjective Optimization*, Boston, USA: Kluwer Academic Publishers, 1999.

[16] J. J. Liang, C. T. Yue, B. Y. Qu. Multimodal multi-objective optimization: A preliminary study. In *Proceedings of IEEE Congress on Evolutionary Computation*, IEEE, Vancouver, Canada, pp. 2454–2461, 2016. DOI: 10.1109/CEC.2016.7744093.

[17] S. C. Maree, T. Alderliesten, P. A. N. Bosman. Real-valued evolutionary multi-modal multi-objective optimization by hill-valley clustering. In *Proceedings of the Genetic and Evolutionary Computation Conference*, Association for Computing Machinery, Prague, Czech Republic, pp. 568–576, 2019. DOI: 10.1145/3321707.3321759.

[18] C. T. Yue, B. Y. Qu, J. Liang. A multi-objective particle swarm optimizer using ring topology for solving multimodal multi-objective problems. *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 5, pp. 805–817, 2018. DOI: 10.1109/TEVC.2017.2754271.

[19] R. Tanabe, H. Ishibuchi. A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In *Proceedings of the 15th International Conference on Parallel Problem Solving from Nature*, Springer, Coimbra, Portugal, pp. 249–261, 2018. DOI: 10.1007/978-3-319-99253-2_20.

[20] R. Tanabe, H. Ishibuchi. A framework to handle multimodal multiobjective optimization in decomposition-based evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 4, pp. 720–734, 2020. DOI: 10.1109/TEVC.2019.2949841.

[21] Y. M. Peng, H. Ishibuchi. A decomposition-based large-scale multi-modal multi-objective optimization algorithm. In *Proceedings of IEEE Congress on Evolutionary Computation*, IEEE, Glasgow, UK, pp. 1–8, 2020. DOI: 10.1109/CEC48606.2020.9185674.

[22] R. Tanabe, H. Ishibuchi. A niching indicator-based multi-modal many-objective optimizer. *Swarm and Evolutionary Computation*, vol. 49, pp. 134–146, 2019. DOI: 10.1016/j.swevo.2019.06.001.

[23] Q. F. Zhang, W. D. Liu, H. Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Proceedings of IEEE Congress on Evolutionary Computation*, IEEE, Trondheim, Norway,

pp. 203–208, 2009. DOI: 10.1109/CEC.2009.4982949.

[24] Q. Kang, X. Y. Song, M. C. Zhou, L. Li. A collaborative resource allocation strategy for decomposition-based multiobjective evolutionary algorithms. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2416–2423, 2019. DOI: 10.1109/TSMC.2018.2818175.

[25] A. M. Zhou, Q. F. Zhang. Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016. DOI: 10.1109/TEVC.2015.2424251.

[26] Q. Z. Lin, G. M. Jin, Y. P. Ma, K. C. Wong, C. A. Coello Coello, J. Q. Li, J. Y. Chen, J. Zhang. A diversity-enhanced resource allocation strategy for decomposition-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2388–2401, 2018. DOI: 10.1109/TCYB.2017.2739185.

[27] X. Y. Cai, Y. X. Li, Z. Fan, Q. F. Zhang. An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 508–523, 2015. DOI: 10.1109/TEVC.2014.2350995.

[28] Y. Xiang, Y. R. Zhou, L. P. Tang, Z. F. Chen. A decomposition-based many-objective artificial bee colony algorithm. *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 287–300, 2019. DOI: 10.1109/TCYB.2017.2772250.

[29] K. Li, A. Fialho, S. Kwong, Q. F. Zhang. Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 1, pp. 114–130, 2014. DOI: 10.1109/TEVC.2013.2239648.

[30] H. L. Liu, L. Chen, Q. F. Zhang, K. Deb. Adaptively allocating search effort in challenging many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 433–448, 2018. DOI: 10.1109/TEVC.2017.2725902.

[31] H. K. Chen, G. H. Wu, W. Pedrycz, P. N. Suganthan, L. N. Xing, X. M. Zhu. An adaptive resource allocation strategy for objective space partition-based multiobjective optimization. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 3, pp. 1507–1522, 2021. DOI: 10.1109/TSMC.2019.2898456.

[32] J. J. Zhou, L. Gao, X. Y. Li. Ensemble of dynamic resource allocation strategies for decomposition-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 710–723, 2021. DOI: 10.1109/TEVC.2021.3060899.

[33] F. van den Bergh, A. P. Engelbrecht. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004. DOI: 10.1109/TEVC.2004.826069.

[34] Y. J. Shi, H. F. Teng, Z. Q. Li. Cooperative co-evolutionary differential evolution for function optimization. In *Proceedings of the 1st International Conference on Advances in Natural Computation*, Springer, Changsha, China, pp. 1080–1088, 2005. DOI: 10.1007/11539117_147.

[35] J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006. DOI: 10.1109/TEVC.2005.857610.

[36] Z. Y. Yang, K. Tang, X. Yao. Large scale evolutionary optimization using cooperative coevolution. *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008. DOI: 10.1016/j.ins.2008.02.017.

[37] M. Yang, M. N. Omidvar, C. H. Li, X. D. Li, Z. H. Cai, B. Kazimipour, X. Yao. Efficient resource allocation in cooperative co-evolution for large-scale global optimization. *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 4, pp. 493–505, 2017. DOI: 10.1109/TEVC.2016.2627581.

[38] Y. H. Jia, W. N. Chen, T. L. Gu, H. X. Zhang, H. Q. Yuan, S. Kwong, J. Zhang. Distributed cooperative co-evolution with adaptive computing resource allocation for large scale optimization. *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 188–202, 2019. DOI: 10.1109/TEVC.2018.2817889.

[39] X. N. Shen, Y. N. Guo, A. M. Li. Cooperative coevolution with an improved resource allocation for large-scale multiobjective software project scheduling. *Applied Soft Computing*, vol. 88, Article number 106059, 2020. DOI: 10.1016/j.asoc.2019.106059.

[40] Y. H. Jia, Y. Mei, M. J. Zhang. Contribution-based cooperative co-evolution for nonseparable large-scale problems with overlapping subcomponents. *IEEE Transactions on Cybernetics*, to be published. DOI: 10.1109/TCYB.2020.3025577.

[41] M. G. Gong, Z. D. Tang, H. Li, J. Zhang. Evolutionary multitasking with dynamic resource allocating strategy. *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 858–869, 2019. DOI: 10.1109/TEVC.2019.2893614.

[42] Y. Wang, H. C. Tan, Y. K. Wu, J. K. Peng. Hybrid electric vehicle energy management with computer vision and deep reinforcement learning. *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 3857–3868, 2021. DOI: 10.1109/TII.2020.3015748.

[43] Y. Deng, F. Bao, Y. Y. Kong, Z. Q. Ren, Q. H. Dai. Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2017. DOI: 10.1109/TNNLS.2016.2522401.

[44] Y. X. Wang, K. Wang, H. W. Huang, T. M. Miyazaki, S. Guo. Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications. *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 976–986, 2019. DOI: 10.1109/TII.2018.2883991.

[45] W. Y. Wang, J. W. Li, X. D. He. Deep reinforcement learning for NLP. in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, Association for Computational Linguistics, Melbourne, Australia, pp. 19–21, 2018. DOI: 10.18653/v1/P18-5007.

[46] Z. Q. Wan, C. Jiang, M. Fahad, Z. Ni, Y. Guo, H. B. He. Robot-assisted pedestrian regulation based on deep reinforcement learning. *IEEE Transactions on Cybernetics*, vol. 50, no. 4, pp. 1669–1682, 2020. DOI: 10.1109/TCYB.2018.2878977.

[47] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, J. G. Slootweg. On-line building energy optimization using deep reinforcement learning. *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3698–3708, 2019. DOI: 10.1109/TSG.2018.2834219.

[48] X. Y. Zhang, Y. Tian, Y. C. Jin. A knee point-driven evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6,

pp. 761–776, 2015. DOI: 10.1109/TEVC.2014.2378512.

[49] K. X. Wei, A. Aviles-Rivero, J. W. Liang, Y. Fu, C. B. Schonlieb, H. Huang. Tuning-free plug-and-play proximal algorithm for inverse imaging problems. In *Proceedings of the 37th International Conference on Machine Learning*, ICML, Vienna, Austria, pp. 10158–10169, 2020.

[50] J. Liang, B. Y. Qu, D. W. Gong, C. T. Yue. Problem definitions and evaluation criteria for the CEC 2019 special session on multimodal multiobjective optimization. Technical Report, Zhengzhou University, China, 2019. DOI: 10.13140 RG.2.2.33423.64164.

[51] E. Zitzler, L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999. DOI: 10.1109/4235.797969.

[52] S. Bandyopadhyay, S. K. Pal, B. Aruna. Multiobjective GAs, quantitative indices, and pattern classification. *IEEE Transactions on Systems*, *Man*, *and Cybernetics* – *Part B*: *Cybernetics*, vol. 34, no. 5, pp. 2088–2099, 2004. DOI: 10.1109/TSMCB.2004.834438.

[53] B. Y. Qu, C. Li, J. Liang, L. Yan, K. J. Yu, Y. S. Zhu. A self-organized speciation based multi-objective particle swarm optimizer for multimodal multi-objective problems. *Applied Soft Computing*, vol. 86, Article number 105886, 2020. DOI: 10.1016/j.asoc.2019.105886.

[54] M. Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937. DOI: 10.1080/01621459.1937.10503522.

[55] Y. Liu, H. Ishibuchi, G. G. Yen, Y. Nojima, N. Masuyama. Handling imbalance between convergence and diversity in the decision space in evolutionary multi-modal multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 3, pp. 551–565, 2020. DOI: 10.1109/TEVC.2019.2938557.

[56] Y. P. Liu, H. Ishibuchi, Y. Nojima, N. Masuyama, K. Shang. A double-niched evolutionary algorithm and its behavior on polygon-based problems. In *Proceedings of the 15th International Conference on Parallel Problem Solving from Nature*, Springer, Coimbra, Portugal, pp. 262–273, 2018. DOI: 10.1007/978-3-319-99253-2_21.

[57] Y. P. Liu, G. G. Yen, D. W. Gong. A multimodal multiobjective evolutionary algorithm using two-archive and recombination strategies. *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 660–674, 2019. DOI: 10.1109/TEVC.2018.2879406.

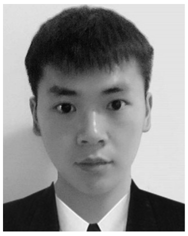
[58] K. Deb, S. Tiwari. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008. DOI: 10.1016/j.ejor.2006.06.042.

[59] A. M. Zhou, Q. F. Zhang, Y. C. Jin. Approximating the set of Pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1167–1189, 2009. DOI: 10.1109/TEVC.2009.2021467.

Qian-Long Dang received the B. Eng. degree in applied mathematics from Henan University of Technology, China in 2016. He is currently a Ph. D. degree candidate in School of Mathematics and Statistics, Xidian University, China.

His research interests include computational intelligence, swarm intelligence, evolution algorithm, and their applications.

E-mail: xidianqldang@163.com (Corresponding author)
ORCID iD: 0000-0001-9295-1361



Wei Xu received the B. Eng. degree in applied mathematics from Inner Mongolia University of Science and Technology, China in 2018. He is currently a master student in School of Mathematics and Statistics, Xidian University, China.

His research interests include computational intelligence, swarm intelligence, evolution algorithm, and their applications.

E-mail: xuwei18782149972@163.com

Yang-Fei Yuan received the B. Eng. degree in applied mathematics from Inner Mongolia University of Science and Technology, China in 2018. He is currently a master student in School of Mathematics and Statistics, Xidian University, China.

His research interests include computational intelligence, swarm intelligence, evolution algorithm, and their applications.

E-mail: st123456yyf@163.com

Machine Intelligence Research

Citation: Q. L. Dang, W. Xu, Y. F. Yuan. A dynamic resource allocation strategy with reinforcement learning for multimodal multi–objective optimization. *Machine Inteligence Research*, vol.19, no.2, pp.138–152, 2022. https://doi.org/10.1007/s11633–022–1314–7

Articles may interest you

Satellite integration into 5g: deep reinforcement learning for network selection. *Machine Intelligence Research*, vol.19, no.2, pp.127-137, 2022.
DOI: 10.1007/s11633-022-1326-3

A framework for distributed semi-supervised learning using single-layer feedforward networks. *Machine Intelligence Research*, vol.19, no.1, pp.63-74, 2022.
DOI: 10.1007/s11633-022-1315-6

Evolutionary computation for expensive optimization: a survey. *Machine Intelligence Research*, vol.19, no.1, pp.3-23, 2022.
DOI: 10.1007/s11633-022-1317-4

Continuous-time distributed heavy-ball algorithm for distributed convex optimization over undirected and directed graphs. *Machine Intelligence Research*, vol.19, no.1, pp.75-88, 2022.
DOI: 10.1007/s11633-022-1319-2

Knowledge mining: a cross-disciplinary survey. *Machine Intelligence Research*, vol.19, no.2, pp.89-114, 2022.
DOI: 10.1007/s11633-022-1323-6

Multi-dimensional classification via selective feature augmentation. *Machine Intelligence Research*, vol.19, no.1, pp.38-51, 2022.
DOI: 10.1007/s11633-022-1316-5

Sharing weights in shallow layers via rotation group equivariant convolutions. *Machine Intelligence Research*, vol.19, no.2, pp.115-126, 2022.
DOI: 10.1007/s11633-022-1324-5