Computers & Industrial Engineering 169 (2022) 108145

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Multimodal multi-objective evolutionary algorithm for multiple path planning

Xingyi Yao [a,1], Wenhua Li [a,1], Xiaogang Pan [a], Rui Wang [a,b,*]

[a] College of Systems Engineering, National University of Defense Technology, Changsha 410073, China
[b] Hunan Key Laboratory of Multi-energy System Intel ligent Interconnection Technology, Changsha 410073, China

ARTICLE INFO

Keywords:
Evolutionary algorithm
Multiple Path planning
Multi-modal
Discrete optimization
Multi-objective optimization

ABSTRACT

The multi-objective path planning problem has received much attention recently. Traditional solving methods try to find a single optimal path without considering the multiformity of the paths. In this study, we first analyze the situation that several different paths may have the same objective values, termed as multi-modal minimum path problems. To address these problems, we propose a novel solution-encoding method, which decreases the size of decision-space greatly. Then, to maintain the population diversity in the decision space, we propose an environmental selection strategy, in which the duplicate solutions are deleted first and then a second-selection method is adopted. Finally, an effective multi-objective evolutionary algorithm based on the special environmental selection is proposed, termed MMEA-SES. Through the experiments, the proposed method is proved effective and efficient compared to other state-of-the-art algorithms for multimodal multi-objective path planning.

1. Introduction

Many problems in the real world require optimizing multiple objectives simultaneously, termed multiobjective optimization problems (MOPs). Generally, the objectives are often in conflict with one another; that is, an improvement in one objective will lead to degradation in at least one other objective (Deb & Agrawal, 1995; Li, Wang, Zhang, Ming, & Li, 2020). In general, an MOP can be expressed as:

$$\begin{aligned}&\text{Minimize } F(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})\},\\ &s.t. \quad \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \Omega,\end{aligned} \tag{1}$$

where Ω denotes the search space, m is the number of objectives, and $\mathbf{x}$ is a decision vector that consists of n decision variables x_i. A solution $\mathbf{x_a}$ is considered to Pareto dominate another solution $\mathbf{x_b}$ *iff* $\forall i = 1, 2, \ldots, m$, $f_i(\mathbf{x_a}) \leqslant f_i(\mathbf{x_b})$ and $\exists j = 1, 2, \ldots, m, f_j(\mathbf{x_a}) < f_j(\mathbf{x_b})$. Furthermore, a Pareto optimal solution is a solution that is not Pareto dominated by any other solution. The set of Pareto optimal solution is called Pareto set (PS). The image of PS is known as Pareto Front (PF).

To address MOPs, many multi-objective evolutionary algorithms (MOEAs) have been proposed and have been proved effective (Li, Wang, Zhang, & Ishibuchi, 2018). Generally speaking, MOEAs can be roughly divided into three categories: Pareto-dominance-based MOEAs (Deb, Pratap, Agarwal, & Meyarivan, 2002), indicator-based MOEAs (Li, Zhang, Wang, & Ishibuchi, 2021; Zitzler & Künzli, 2004) and MOEAs based on de-composition (Wang, Ishibuchi, Zhou, Liao, & Zhang, 2018). The main target of these MOEAs is to approximate the Pareto front with good convergence quality and well distribution in the objective space. Over the last decades, many proposed MOEAs show good performances on benchmark problems in terms of the proposed metrics like IGD (Ishibuchi, Imada, Setoguchi, & Nojima, 2018). For most of them, the diversity in the objective space is considered to obtain a well-distributed PF.

Generally, maintaining the solution diversity in the decision space while seeking the convergence quality in the objective space is hard to balance in a single algorithm run (Liu, Ishibuchi, Yen, Nojima, & Masuyama, 2019). There arises a situation in real-world MOPs that, there may exist two or more global or local PSs and some of them may correspond to the same PF. These problems are defined as multimodal multi-objective optimization problems (MMOPs) (Li, Zhang, Wang, et al., 2021; Liang et al., 2019; Li, Yao, Zhang, Wang, & Wang, 2022), e. g., for example, diet design problems (Rudolph, Naujoks, & Preuss, 2007), space mission design problems (Schütze, Vasile, & Coello Coello,

* Corresponding author at: College of Systems Engineering, National University of Defense Technology, Changsha 410073, China.
E-mail address: ruiwangnudt@gmail.com (R. Wang).

[1] Xingyi Yao and Wenhua Li have equal contributions to this work and should be considered as co-first authors.

https://doi.org/10.1016/j.cie.2022.108145
Received 10 December 2021; Received in revised form 27 March 2022; Accepted 31 March 2022
Available online 15 April 2022
0360-8352/

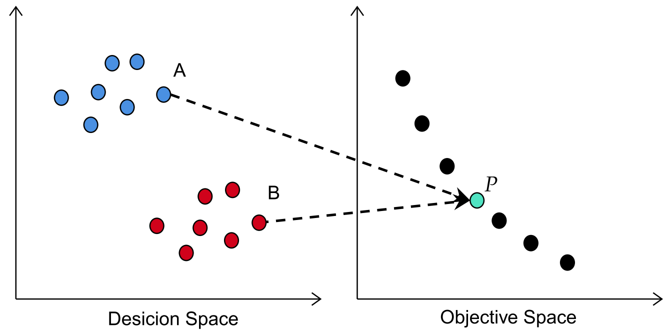

Fig. 1. Explanation of MMOP.

2011), rocket engine design problems (Kudo, Yoshikawa, & Furuhashi, 2011), and functional brain imaging problems (Sebag, Tarrisson, Teytaud, Lefevre, & Baillet, 2005). Fig. 1 is an explanation of MMOP, in which point A and point B are distant in the decision space and they share the same objective vector P.

As we can see from Fig. 1, for real-world engineering problems, both solutions A and B are needed to obtain. On the one hand, the decision-makers (DMs) can better understand the problems that need to be optimized; on the other hand, if one certain solution becomes infeasible due to the environmental change, a DM can easily change to another equivalent solution. Therefore, obtaining all equivalent PSs is important for some problems. To solve these MMOPs, many multi-modal multi-objective evolutionary algorithms (MMEAs) have been proposed. For example, omni-optimizer (Deb & Tiwari, 2005, 2008), DNEA (Liu, Ishibuchi, Nojima, Masuyama, & Shang, 2018), MMEA-WI (Li, Zhang, Wang, et al., 2021), CPDEA (Liu et al., 2019) and so on. However, they are designed especially for continuous problems. For now, the performance of these proposed MMEAs on solving discrete problems has not been studied.

Path planning problems have been studied for several years (Huang, Yuan, Shi, Liu, & Wu, 2021; Siddiqi, Milani, Jazar, & Marzbani, 2021). Among them, the optimal path is always the first target. However, in the travel path planning problems, users want to know several paths that have the same objective values. Moreover, some users may like to accept other paths that are a little worse than the optimal solution. In this case, improving the diversity of the obtained solutions is of significant importance. To address this problem, this study first explains the importance of utilizing multi-modal multi-objective algorithms to solve path planning problems. Then, we proposed a novel MMEA based on special environmental selection to maintain the solution diversity in the decision space. Finally, several experiments are used to prove the effectiveness of our proposed algorithm. In conclusion, the contributions of this work can be concluded as:

- The differences between discrete and continuous multimodal multi-objective optimization problems are analyzed. The existing MMEAs are shown to be not suitable to solve discrete problems due to the difficulty of defining the solution distances.
- A novel crowding distance calculation method is proposed for multimodal multi-objective path planning problems. Based on the proposed method, a diversity-based fitness evaluation method is proposed.
- A special environmental selection strategy is proposed to maintain the diversity as well as the convergence quality of solutions, resulting in a novel multimodal multi-objective evolutionary algorithm, termed MMEA-SES.
- Several experiments have been conducted, which show that the proposed algorithm is competitive compared to other state-of-the-art algorithms. In addition, the effect of the special environmental selection strategy is further analyzed.

The rest of this study is structured as follows: Section 2 illustrates the path planning problem we studied in this paper and the recent works about multimodal multi-objective optimization, followed by Section 3, which gives a detailed explanation of the proposed evolutionary algorithm. Then, in Section 4, we design several experiments to prove the effectiveness of the proposed method. Finally, the conclusion of the work and the possible future work are given in Section 5.

2. Preliminary work

2.1. Multimodal multi-objective path planning

Path planning problems are representative MMOPs. However, few works try to find all optimal solutions for these kinds of problems. In the multimodal multiobjective path planning (MMOPP) test suite of CEC 2021 (Liang et al., 2020), the authors proposed a test suite of multimodal path planning problems. Specifically, Fig. 2 shows the express network and its simulated map of Beijing. From Fig. 2, we can see that black areas are feasible paths, white areas are infeasible regions and red areas are congested regions. Moreover, the blue circles are intersections,

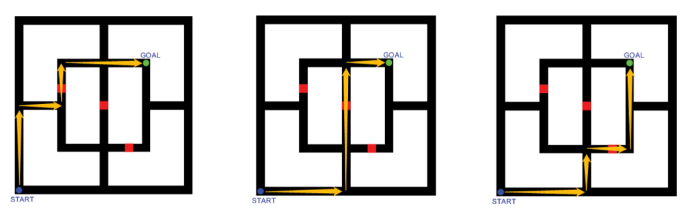

Fig. 3. Three optimal paths which have exactly the same path length and number of red points.

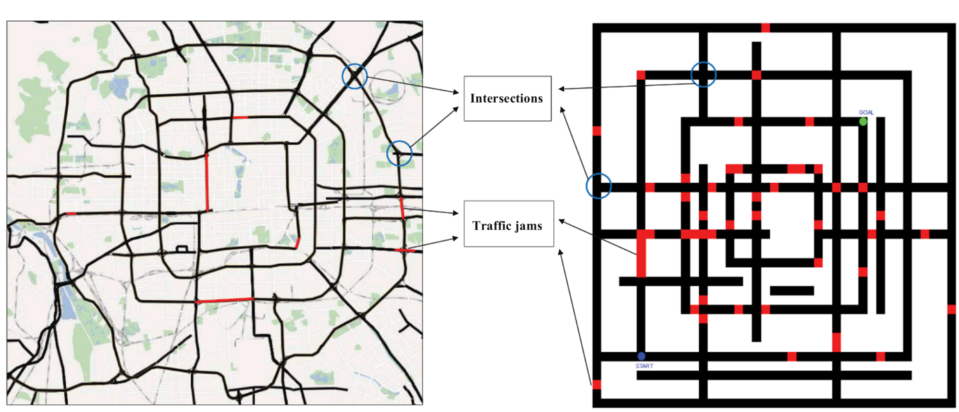

Fig. 2. Beijing express network and the simulated map (Liang et al., 2020), where the red lines represent traffic jams and blue circles are intersections, respectively.

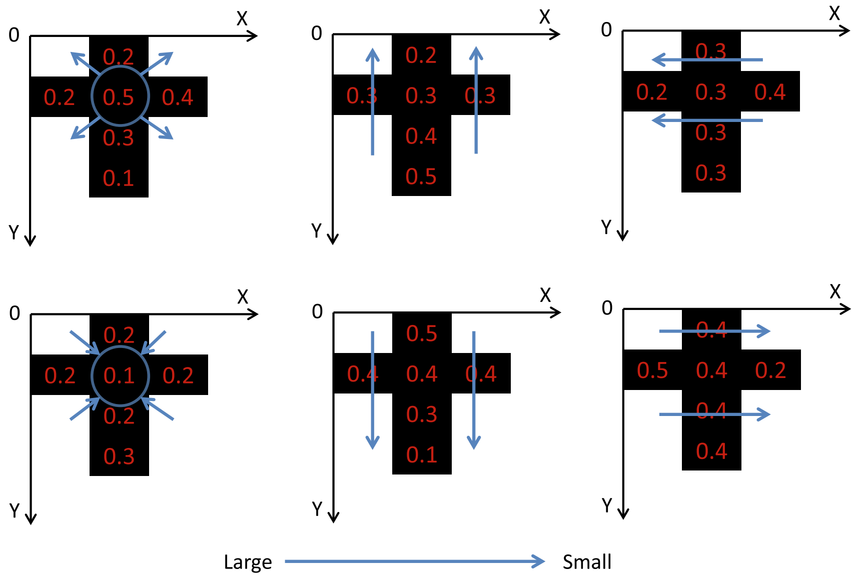

Fig. 4. The values assigned to each point for the second type problems.

which also mean traffic lights. Reasonably, the optimal paths should have the shortest path length, the minimum number of intersections, and the minimum number of congested regions.

Arguably, finding one of them may be sufficient to obtain an acceptable solution. However, failing to identify more than one of the shortest paths may prevent the decision-maker from considering solution options that could bring about improved performance. Different decision-makers have different preferences, so providing multiple excellent solutions is necessary.

As we can see from Fig. 3, three different paths have the same objective values, namely, the path length and the number of red points. To give deeper research on these problems, the CEC 2021 test suit is proposed which can be roughly categorized into three different types. For the first type, there are three objectives needing to be minimized, namely, the path length, the number of intersections, and the number of the red points, which can be also observed in Fig. 2. For the second, the authors consider that using red points to represent the congestions is inaccurate. Thus, for each feasible point, values are assigned to represent the degree of crowdedness or road width. Fig. 4 shows the values assigned to points. It's worth mentioning that, more than one real number can be assigned to the point in the map. Thus, except for the first objective (path length), there could be more than one cumulative value, see Fig. 4. As a result, for the second type of problem, the number of objectives is up to 7. According to the paper (Liang et al., 2020), the cumulative value F has several pattern. That is, from the outside to the inside, F are getting larger; from the outside to the inside, F are getting smaller; from the up to the down, F are getting larger; from the up to the down, F are getting smaller; from the left to the right, F are getting larger; from the left to the right, F are getting smaller. For the third type, the authors consider that path planning would need to consider the point that must be accessed, which is marked by the yellow points.

2.2. Multimodal multi-objective optimization

As we mentioned above, MMOPs are a kind of MOP in which multiple Pareto optimal solution sets correspond to equivalent Pareto fronts (Li, Zhang, Wang, et al., 2021). In order to better understand, we adopt the definition of MMOPs given by Rudolph and Preuss in Rudolph and Preuss (2009), which can be described as follows:

Definition 1. An MMOP involves finding all solutions that are equivalent to Pareto optimal solutions.

Definition 2. Two different solutions $\mathbf{x_1}$ and $\mathbf{x_2}$ are said to be equivalent if and only if $\|F(\mathbf{x_1}) - F(\mathbf{x_2})\| \leqslant \sigma$, where $\|\mathbf{a}\|$ is an arbitrary norm of $\mathbf{a}$.

In this study, we only consider the case $\sigma = 0$. Specifically, an MMOP is an MOP that has multiple Pareto solutions sets corresponding to the same Pareto front.

Over the proposed MMEAs, the omni-optimizer (Deb & Tiwari, 2005, 2008) is perhaps the most representative one. In the algorithm, a new crowding distance is proposed, which takes the solution diversity in both the objective and decision spaces into account. Specifically, the crowding distance $c_{i,j}^{\text{dec}}$ in the objective space is calculated as follows.

$$c_{i,j}^{\text{dec}} = \begin{cases} 2\left(\dfrac{x_{i+1,j} - x_{i,j}}{x_j^{\max} - x_j^{\min}}\right) & \text{if } x_{i,j} = x_j^{\min}, \\ 2\left(\dfrac{x_{i,j} - x_{i-1,j}}{x_j^{\max} - x_j^{\min}}\right) & \text{else if } x_{i,j} = x_j^{\max}, \\ \dfrac{x_{i+1,j} - x_{i-1,j}}{x_j^{\max} - x_j^{\min}} & \text{otherwise}, \end{cases} \tag{2}$$

where $x_j^{\max}$ and $x_j^{\min}$ are the maximum and minimum values of the j-th variable, respectively.

Motivated by omni-optimizer, other Pareto-dominance-based MMEAs adopt different diversity-maintenance strategies, e.g., DNEA (Liu et al., 2018) and DN-NSGA-II (Liang, Yue, & Qu, 2016). Notably, the particle swarm optimization (PSO) variation operator have advantages in MMOPs. MO_PSO_MM (Liang, Guo, Yue, Qu, & Yu, 2018) and MO_Ring_PSO_SCD (Yue, Qu, & Liang, 2017) are proposed with the special crowding distances to maintain the diversity of solutions in the decision space.

In addition, MOEA/D-AD (Tanabe & Ishibuchi, 2018) assigns one or more individuals to each subproblem to handle multimodality. Such strategies can also be found in Hu and Ishibuchi (2018). MMODE (Liang et al., 2019) introduces a modified differential evolution (DE) operator to promote diversity in the decision space. In addition, Tanabe and Ishibuchi (2019) proposed a niching indicator-based multimodal many-objective optimizer (NIMMO). Very recently, a weighted indicator-based evolutionary algorithm for multimodal multi-objective optimization (MMEA-WI) (Li, Zhang, Wang, et al., 2021) is proposed to solve the MMOPs. The weighted indicator I_i^w can be expressed as:

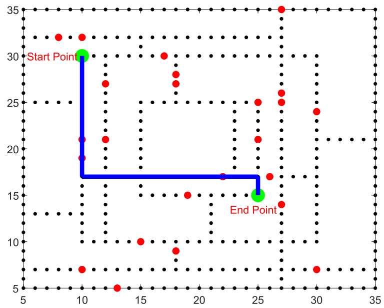

Fig. 5. Illustration of the encoding method and a passable path of Problem 1.

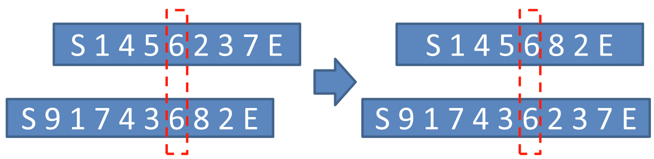

Fig. 6. Illustration of crossover operation.

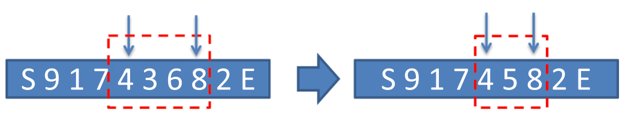

Fig. 7. Illustration of mutation operation.

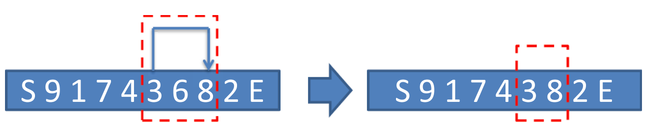

Fig. 8. Illustration for a solution-modified method.

$$I_i^w = \sum_{j=1}^{N} w_{i,j} F(x^j), \; i = 1, 2, \ldots, N, \tag{3}$$

where N and $w_{i,j}$ are the population size and the weighted vector of solution x_i. I_i^w is the weighted sum of fitness $F(x^j)$ with weights $w_{i,j}$. The weight $w_{i,j}$ is calculated according to the Euclidean distance between solutions i and j in the decision space. Through the transformation, solutions located exactly on one of the true PSs may have the same fitness values as solutions located near the same true PS. Therefore, diversity in the decision space can be well maintained.

To our knowledge, although the above-mentioned MMEAs have been widely proved effective in solving benchmark MMOPs, the effect on discrete problems have not been studied yet. In addition, the basic idea of these MMEAs is to select solutions based on the solution distance in the decision space, which is hard to define for discrete optimization problems.

3. Evolutionary algorithm with special environmental selection

3.1. Motivation and the general framework

MOEAs have been widely studied. Traditional MOEAs try to obtain solutions with better convergence quality (Li, Wang, Zhang, Ming, & Lei, 2019). Thus, they adopt the convergence-first strategy to update the population. In other words, a solution with better convergence will more likely be selected to form the new population. However, for problems with more than one different solution corresponding to the same objective values, the convergence-first strategy will lead to quick premature and the lack of diversity-maintenance mechanism will cause poor quality in finding all equivalent solutions. The existing MMEAs focus mainly on continuous optimization problems. Since they need to calculate the distances between solutions in the decision space, it's hard to perform these MMEAs to solve multimodal path planning problems.

In this study, we proposed a novel multi-objective evolutionary algorithm with special environmental selection to obtain as many optimal solutions as many as possible, termed MMEA-SES. Generally, we use the framework shown in Algorithm 1. As we can see, the general framework of MMEA-SES is similar to that of NSGA-II (Deb et al., 2002). Similar to most MOEAs, MMEA-SES consists of the following parts: population initialization, mating selection, offspring generation and environmental selection. Notably, during the evolution process, we will first use the tournament selection method to select the parents based on the proposed diversity-based fitness f^d to produce the offspring, see *lines* 3 −4 in Algorithm 1. After that, the special environmental selection method is executed to produce the new population and the diversity-based fitness f^d for next-generation, *line* 5. Notably, the special environmental selection method will be further illustrated in the following section.

Algorithm 1. General Framework of MMEA-SES

Input: Maximum generations *MaxGen*, population size *N*
Output: Optimal solutions *Arc*
1: $Pop \leftarrow Initialization(N)$
2: **while** $gen \leqslant MaxGen$ **do**
3: $MatingPool \leftarrow TournamentSelection(Pop, f^d)$
4: $Off \leftarrow Variation(MatingPool)$
5: $[Pop, f^d] \leftarrow EnvironmentalSelection(Pop, Off, N)$
6: $gen \leftarrow gen + 1$
7: **end while**
8: $Arc \leftarrow UpdateArc(Pop)$

3.2. Encoding method

The main problem of utilizing MOEAs to solve real-world engineering problems is to correctly encode a solution (Li, Zhang, Zhang, & Huang, 2020; Li, Zhang, Yang, Tao, & Xu, 2021). Since the path planning problem is discrete, we use a sequence to represent a path. As we can see from Fig. 5, we first find all the turn points from the map and numbered them. Then, a path from the start point to the endpoint can be presented as a string of numbers. For example, for problem 1, there are 45 turn points, as we can see from Fig. 5. The start point and endpoint are numbered as 11 and 28 respectively. Then a passable path (contains 30 points) is represented as 11→9→29→28, shown as Fig. 5. With this coding method, we can decrease the size of the decision space greatly, which can help accelerate the searching process.

To generate the initial population, we first find all the reachable turn points for each turn point and save them as lists. Then, beginning with the start point, we will randomly select the next point from the reachable point list until we reach the endpoint.

3.3. Variation

The crossover and mutation operation is important for improving the searching efficiency. Based on our encoding method, we propose a variation method, which can be simply illustrated in Fig. 6 and Fig. 7. Specifically, for the crossover operation, we first randomly pick two different solutions. Secondly, we will find the common pass points, e.g., point 6 in Fig. 6. Then, we exchange the following passed points after this common point. For the mutation operator, we first find two points, e.g., point 4 and point 8 in Fig. 7. Then, we will try to find a new passable way from point 4 to point 8 to replace the original passable way.

Notably, to improve the searching ability of our algorithm, we will fix the solution after each variation. Specifically, for each solution, if the current point can directly reach the next two points, then we will delete

Table 1
Parameters of the test problems and algorithms used in this study.

Problem	Size	Objs	Pop Size	Evaluation
1	40*40	2	300	15000
2	40*40	3	300	15000
3	50*50	3	600	30000
4	50*50	3	600	30000
5	84*84	3	800	200000
6	40*40	2	300	15000
7	40*40	3	300	15000
8	50*50	4	600	50000
9	50*50	5	600	50000
10	84*84	7	5000	1000000
11	40*40	2	300	15000
12	40*40	3	300	15000

Table 2
Computational Complexity of MMEA-SES on 12 Test Problems

Indicator	T1/T0	T2/T0	T3/T0	T4/T0	T5/T0	T6/T0
Value	51.84	96.42	252.55	130.92	1913.46	28.48

Indicator	T7/T0	T8/T0	T9/T0	T10/T0	T11/T0	T12/T0
Value	124.48	389.39	600.39	37087.95	17.96	156.40

the next point, which can be illustrated as Fig. 8. The reason for using this fixing method to modify the solutions is that, after performing the strategy, there will be no retrogression in a certain path. As a result, the fixing method can improve the searching efficiency. However, adopting such a strategy may cause failure in searching paths for the third type of problem, which needs to access the yellow points. To address this issue, we borrow the thought of divide and conquer. Specifically, if there is only one yellow point needs to be visited, then we will run the optimization twice, namely, finding paths from the start point to the yellow point and finding paths from the yellow point to the endpoint. Then, the final solution set is obtained by combining the solutions from two optimizations.

3.4. Environmental selection

There are different solutions with the same objective values. Thus, if we simply apply the convergence-first strategy to select solutions from the joint population, the different solutions with equal objective values may be removed. Naturally, we want to keep the solutions with better Pareto ranking and maintain the diversity of the population in the decision space, which usually conflicts (Liu et al., 2019).

Traditional MOEAs show poor performance in solving MMOPs. The reasons can be concluded as (1) the lack of diversity-maintenance mechanism will lead to quickly convergence and premature, (2) different solutions with the same objective values are difficult to coexist simultaneously. As shown in Fig. 1, we assume that point A has been found in the searching process. For MMEAs, we need to obtain point B as well. However, if a solution moved to point B, then in the objective space, it will move to point P. As a result, this solution is crowded with solution A although they are distant in the decision space. Thus, it's difficult for traditional MOEAs to obtain complete PSs.

To address this issue, we first remove the duplicate solutions (*line* 1 in Algorithm 2). By doing so, we can make sure the next generation has the best solutions found so far and keeps diversity as well. In this stage, traditional MMEAs will first calculate the distance between solutions. Then, the most crowded solution will be deleted. However, most MMEAs are designed especially for continuous optimization problems. For these problems, solutions have the same length of decision variables. Thus, the crowding distance of these solutions can be easily represented by Euclid distance. However, as we mentioned in Section 3.2, we use a novel encoding method to express a solution, which is hard to directly evaluate the crowding distance. To address this issue, we proposed a crowding distance calculation method, which can be expressed as:

$$CD_{i,j} = \frac{2 * L_{i,j}^{same}}{L_i + L_j} \tag{4}$$

where L_i is the path length of solution x_i after decoding, $L_{i,j}^{same}$ means the number of common points in both solutions x_i and x_j after decoding. Specifically, if two solutions are exactly the same, then $CD_{i,j} = 1$. The crowding distance used here can also be considered as the similarity.

After removing the duplicate solutions, we first sort the population by the non-dominated Pareto sorting method, see *lines* $2-4$. If the number of solutions in the joint population is greater than the population size N, a second-selection strategy is performed to keep the population size. Specifically, the second-selection strategy can be divided as (1) calculate the diversity-based fitness f^d; (2) delete the solution with the maximum f^d; (3) repeat steps (1) and (2) until the population size is equal to N.

The diversity-based fitness of solution x_i can be expressed as:

$$f_i^d = \frac{\sum_{j=1}^{N} CD_{i,j} \Big/ CD^{max}}{N} \tag{5}$$

where CD^{max} is the maximum value of $CD_{i,j}$.

Algorithm 2. Environmental selection

Input: Joint population *JointPop*, population size N
Output: Updated population *Pop*, diversity-based fitness f^d
1: $JointPop \leftarrow DelEqualSolutions(JointPop)$
2: $FrontNo \leftarrow NDSort(JointPop)$
3: $Index \leftarrow Sort(FrontNo, 'ascend')$
4: $JointPop \leftarrow JointPop(Index)$
5: **if** $Num(JointPop) > N$ **then**
6: $[Pop, f^d] \leftarrow SecondSelection(JointPop)$
7: **else**
8: $Pop \leftarrow JointPop$
9: **end if**

Table 3
The average number of different solutions obtained by MMEA-SES and the compared algorithms over 31 independent runs.

Problem	NSGA-II	MOEA/D-LWS	MO_PSO_MM	MO_Ring_PSO_SCD	MOASTAR	MACOSX	MMEA-SES	Pareto Set
1	7.59	7.55	8.01	8.31	9	9	9	9
2	15.05	15.89	17.23	18.23	24	22.4	24	24
3	8.15	9.57	11.58	11.43	13	12.65	13	13
4	6.25	7.01	7.82	7.81	9	8.85	9	9
5	5.28	7.32	15.32	14.34	24	17.87	23.43	24
6	4.22	4.15	4.69	4.48	5	4.95	5	5
7	5.26	5.42	13.05	14.32	16	14.23	16	16
8	27.56	28.52	34.62	38.65	48	42.54	48	48
9	55.65	62.68	75.22	84.22	105	101.23	104.88	105
10	368.56	599.53	788.35	818.35	1280	995.3	1275.68	1280
11	3.82	3.65	4	4	4	4	4	4
12	6.03	5.67	13.74	12.5	22	20.13	22	22
Avg Rank	6.3	6	4.3	3.7	1	2.7	1.2	

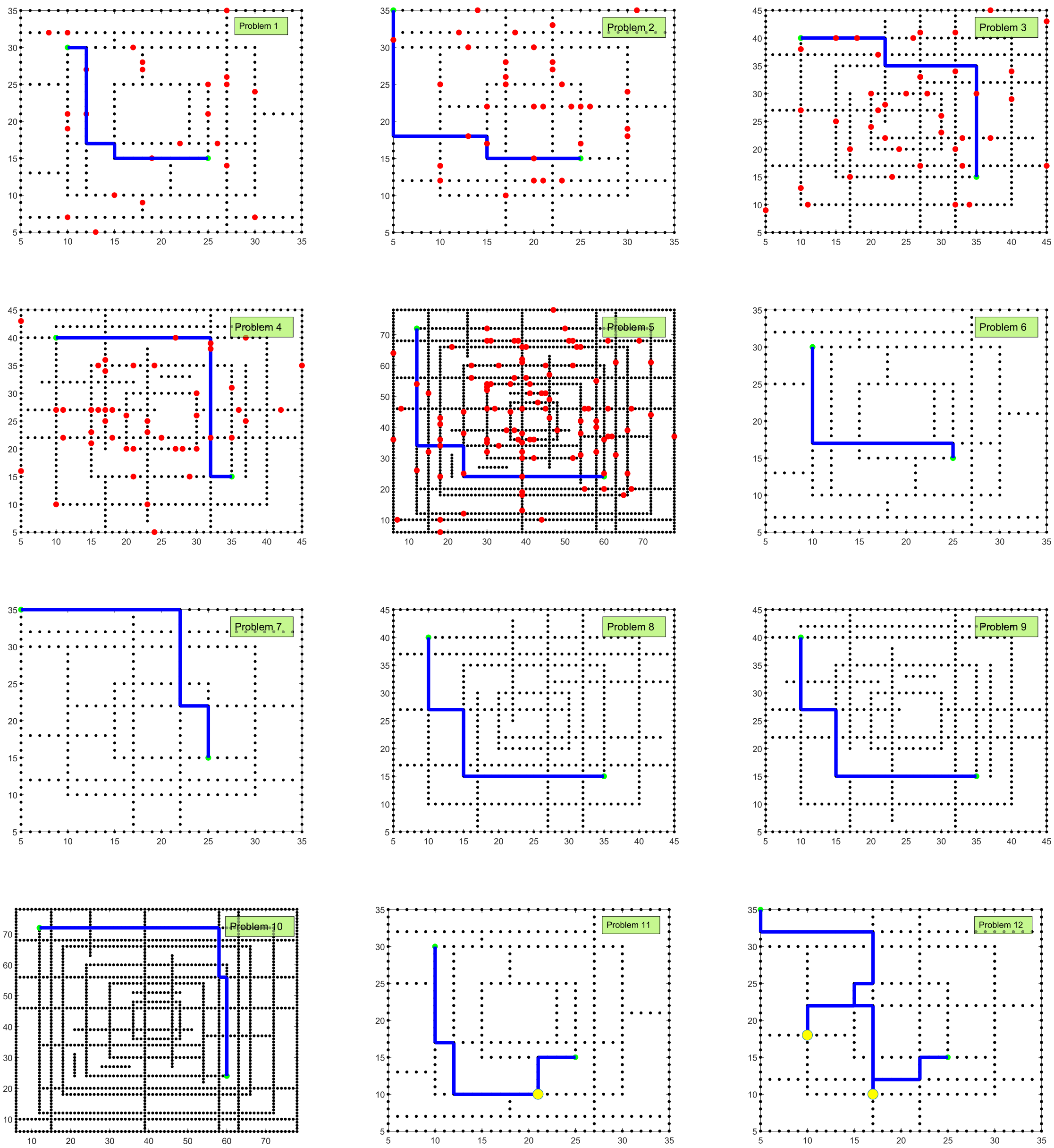

Fig. 9. One of the different optimal solutions for each problem. For Problems 1–5, there are red points representing the traffic jams; for Problems 6–10, the degree of traffic jams is indicated by certain values; for Problems 11–12, the yellow points must be accesses once.

To sum up, as shown in Algorithm 2, we first remove the duplicate solutions to maintain the diversity of solutions in the decision space, *line* 1. Then, to improve the convergence ability of the algorithm, we use the non-dominated sorting method to sort the population, *lines* 2 −4. Finally, the second-selection strategy is performed to maintain the population size, *lines* 5 −9.

Notably, after the evolution, the obtained population is not the final optimal solution set. Thus, we will choose the final solutions by selecting, which can be seen in *line* 8 in Algorithm 1. Specifically, we will select the non-dominated solutions and delete the duplicate solutions.

4. Experiment

4.1. Experimental setting

4.1.1. Compared algorithms and the benchmark problems

To better illustrate the performance of our proposed algorithm, we select several state-of-the-art MOEAs as the competitor algorithms, namely NSGA-II (Deb et al., 2002), MOEA/D-LWS (Wang et al., 2018), MO_PSO_MM (Liang et al., 2018), MO_Ring_PSO_SCD (Yue et al., 2017), MOASTAR (Jin, 2021) and MACOSX (Zhao et al., 2021). Specifically, NSGA-II and MOEA/D-LWS are selected as the representative MOEAs that focus on solving traditional MOPs. MO_PSO_MM and MO_Ring_PSO_SCD are chosen as representative MMEAs that are designed especially for multi-modal problems. Finally, we select MOASTAR and MACOSX as two competitor algorithms that participate in the CEC 2021 competition. Specifically, MOASTAR and MACOSX are the winner and second place algorithms of the CEC 2021 competition. Since there is no suggested running parameter for problems we studied, for all the algorithms, we set the population size N and the maximum number of function evaluations N_E according to Table 1. Note that the specific parameters in each algorithm are set according to the original papers. All the experiments are performed on a PC with Intel i7-1165G7 @ 2.8 GHz (CPU), 16G Ram, and MATLAB 2017a. Notably, for the convenience of the researchers, we have uploaded the source code and made it

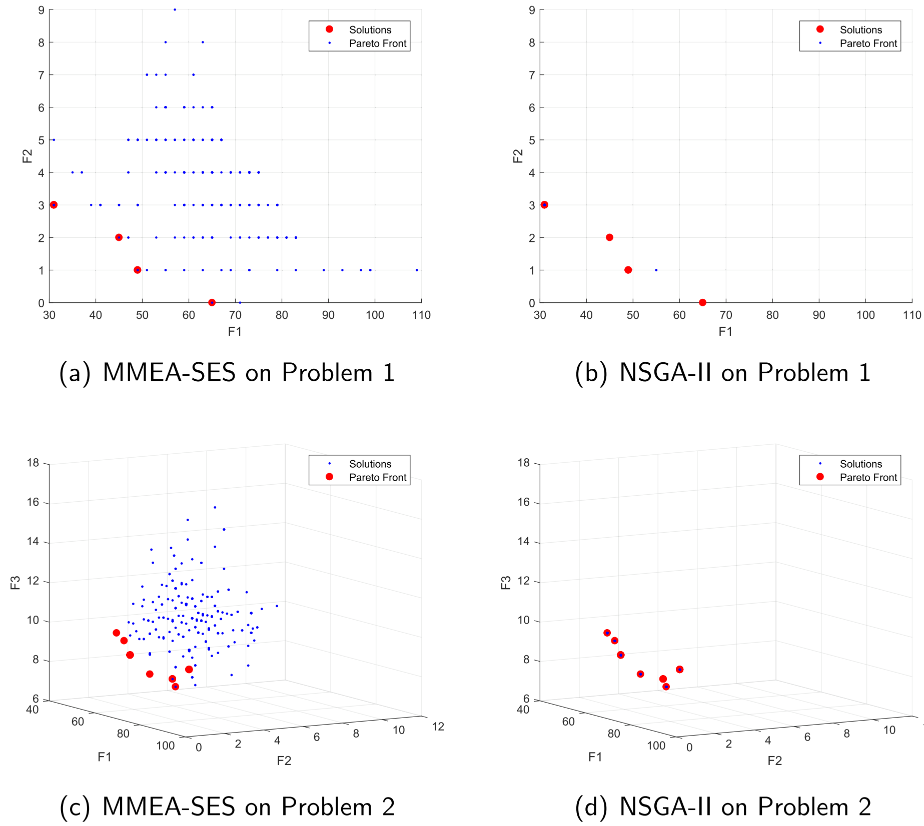

Fig. 10. The Pareto Front and the final population distribution in the objective space obtained by MMEA-SES.

open-access.[2] To reduce the impact of randomness, all the experiments will be executed 31 times. The result closest to the average result will be used to show the performance of all compared algorithms. In addition, the solution that is closest to the average result will be selected to present and analyze. It's worth mentioning that, the true Pareto optimal front is provided from Jin (2021), which is verified by a third-party library named *pareto*.[3] However, the true Pareto optimal set is not given. Thus, we gather all results from the competition and form an un-verified Pareto optimal set, which is also provided in Github.

The parameter setting of the benchmark problems and the algorithm-running can be seen in Table 1. As we can see, the number of objectives is up to 7 for the benchmark problems, which can be used to prove the effectiveness of the algorithms on solving many-objective optimization problems. The population size and the maximum number of function evaluations are set according to the complexity of the problems.

Notably, most of the MMEAs need to calculate the distance between solutions in order to compare the similarity of solutions. However, for problems studied in this paper, the decision space is discrete and the length of solutions is unequal, which makes it hard to calculate the distance. Thus, for other compared algorithms that require calculating the solutions crowding distance, we use Eq. 4 to replace the Euclid distance.

4.1.2. Performance metric

To evaluate the performance of MOEAs, many metrics have been proposed, e.g., the hypervolume, GD, and IGD indicators (Coello & Sierra, 2004; Zitzler, Thiele, Laumanns, Fonseca, & Da Fonseca, 2003). However, for the studied problems in this work, the aim is to find all Pareto optimal solutions which may have the same objective values. As a result, these metrics are inappropriate to show the performance of MOEAs on selecting benchmark problems. Then, in order to evaluate the performance, we use the number of different optimal solutions *NOS* to indicate the performance. Specifically, let S^* and S to be the true Pareto optimal solution set and the obtained solution set, then *NOS* is the number of solutions x^* that satisfies $x^* \in S^*$ and $x^* \in S$.

To evaluate the computational complexity, the organizer proposed to calculate T/T_0, where T_0 represents reference time cost by executing a piece of code and T represents the time cost of the proposed method. Specifically, $T_0 = 0.036s$ for this study. Thus, the average running time complexity of MMEA-SES is shown in Table 2.

4.2. Result obtained by all algorithms

In this section, we select several state-of-the-art MOEAs as the competitor algorithms, namely, NSGA-II, MOEA/D-LWS, MO_PSO_MM, MO_Ring_PSO_SCD, MOASTAR and MACOSX. Then, all the experiments are executed 31 times. Table 3 lists the average number of different solutions *NOS* obtained by all algorithms. Specifically, for Problems 1–12, the numbers of solutions in the true Pareto optimal set are 9, 24, 13, 9, 24, 5, 16, 48, 105, 1280, 4 and 22 respectively. As we can see from Table 3, the average number of MMEA-SES are 9, 24, 13, 9, 23.43, 5, 16, 48, 104.88, 1275.68, 4 and 22 respectively. For most of the benchmark problems, MMEA-SES and MOASTAR can stably obtain all different solutions. However, for Problems 5, 9 and 10, MMEA-SES can not obtain all of the optimal solutions. The gap between the optimal is small. Further analysis shows that this is because the size of the problem is relatively big and the passable paths are complex. In addition, for Problems 9 and 10, the number of objectives is greater than 5, which is hard for MOEAs to obtain the whole PF. As a comparison, MOASTAR performs best over 12 test problems. MMEA-SES perform a bit worse than MOASTAR.

[2] The source code of this study can be found in https://github.com/Wenhua-Li/MMEASES.

[3] The Pareto library is available at https://alandefreitas.github.io/pareto/.

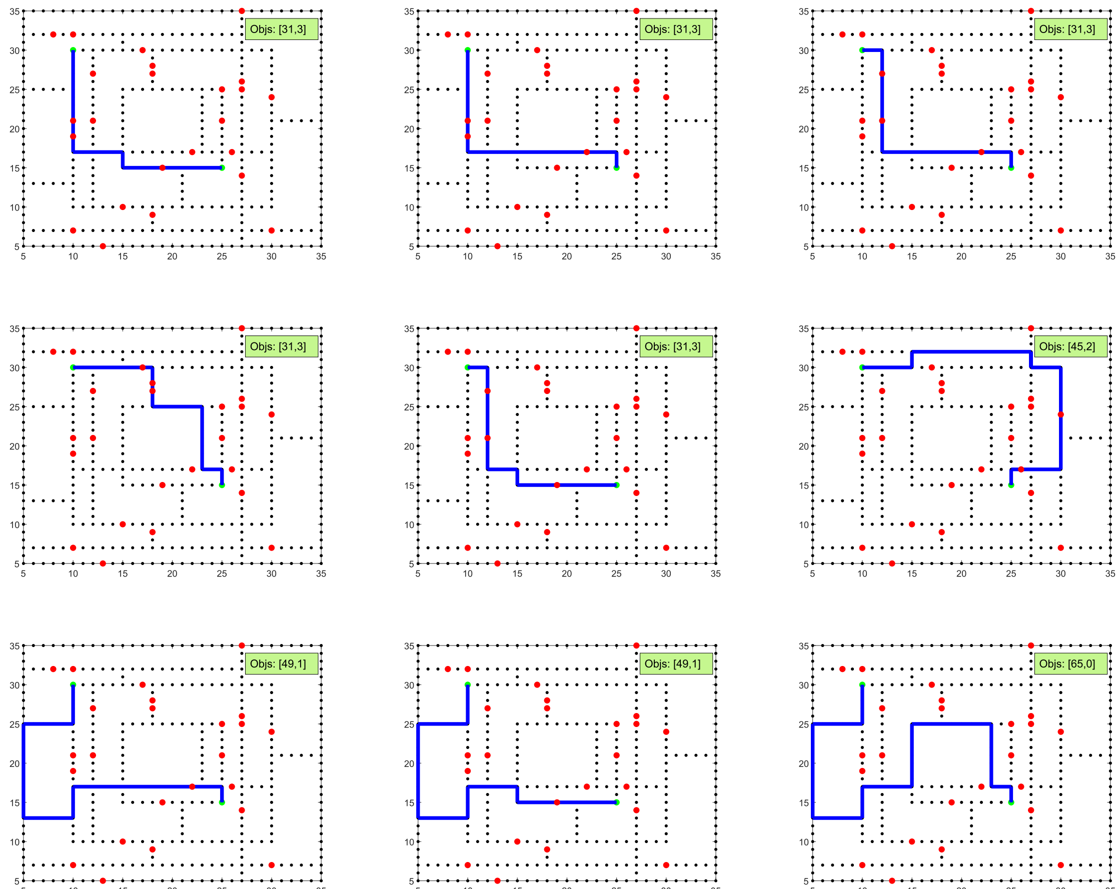

Fig. 11. All different optimal solutions for Problem 1, where the objective vectors are (31,3), (45,2), (49,1) and (65,0) for the 1-5th, 6th, 7-8th, and 9th subfigures, respectively.

Table 4
The average number of different solutions *NOS* obtained by NSGA-II-SES and the compared algorithms over 31 independent runs.

Problem	NSGA-II	NSGA-II-SES	MMEA-SES	True PF
1	7.59	9	9	9
2	15.05	23.02	24	24
3	8.15	12.65	13	13
4	6.25	8.92	9	9
5	5.28	19.51	23.43	24
6	4.22	5	5	5
7	5.26	16	16	16
8	27.56	42.65	48	48
9	55.65	103.25	104.88	105
10	368.56	1152.63	1275.68	1280
11	3.82	4	4	4
12	6.03	22	22	22

Compared to MMEA-SES and MOASTAR, other state-of-the-art algorithms show poor performances on the benchmark problems. Specifically, for traditional MOEAs (NSGA-II and MOEA/D-LWS), MOEA/D-LWS receives the best average rank for these 12 problems. This is because MOEA/D-LWS utilizes the decomposition method to simultaneously optimize the problems. As a result, the uniformity and spread quality of the obtained solutions in the objective space is better than NSGA-II. Thus, the diversity of solutions in the decision space is improved accordingly. Notably, although these three traditional MOEAs are not designed especially for multimodal problems, they show good results on Problems 1, 6 and 11.. For some algorithm runs, they can also obtain all different optimal solutions for these problems. Further analysis shows that this is because the size of these problems is small and the complexity is not high enough.

In addition, MO_PSO_MM and MO_Ring_PSO_SCD are MMEAs that are designed for multimodal multi-objective problems. Thus, it's supposed that MO_PSO_MM and MO_Ring_PSO_SCD can get better results on these problems compared to other traditional MOEAs. According to the results shown in Table 3, MO_PSO_MM and MO_Ring_PSO_SCD show better performance compared to other MOEAs. However, since MO_PSO_MM and MO_Ring_PSO_SCD are not designed for discrete problems, they can not win MOASTAR, MMEA-SES and MACOSX on all of the test problems. As shown in Fig. 9, for each problem, we select one of the optimal different solutions obtained by MMEA-SES to present respectively, from which we can intuitively observe the complexity of each problem.

To sum up, MMEA-SES and MOASTAR perform better than the selected state-of-the-art algorithms on the chosen test problems, while the selected MMEAs have some advantages compared to traditional MOEAs. For these problems, MOASTAR shows its competitiveness by utilizing the A* algorithm (Jin, 2021). Compared to MOASTAR, the proposed MMEA-SES is more generative, which can be easily extended to other discrete optimization problems.

4.3. Further analysis on MMEA-SES

To better illustrate the performance of the proposed MMEA-SES, we present the final phase of population distribution in the objective space of MMEA-SES and NSGA-II. Fig. 10 shows the distribution of solutions in the objective obtained by MMEA-SES and NSGA-II, where the red points are the current Pareto optimal solutions and the blue points are solutions. From Fig. 10 we can see that, for Problem 1 and Problem 2, there are two and three objective functions respectively. According to Table 3, there are 9 and 24 different optimal solutions in total. As we can see in Fig. 10(a) and (c), for MMEA-SES, other dominated solutions are distributed uniformly in the objective space, which shows a great solution diversity. This is because we adopt the special environmental selection strategy. In each generation, the diversity of the population can be well maintained.

Compared to MMEA-SES, the performance of NSGA-II can be intuitively shown in Fig. 10(b) and (d). As we can see, NSGA-II can find part of the PF. That is, the PF obtained by NSGA-II is incomplete. Moreover, other dominated solutions overlap (a single point in the figure can be several solutions). During the evolution of NSGA-II, the population will quickly converge to the optimal and lead to premature. Thus, it's hard for traditional MOEAs to solve multimodal multi-objective problems.

Fig. 11 presents all the optimal solutions for Problem 1 obtained by MMEA-SES. As we can see, the first 5 solutions have the same objective values, namely, (31,3), which means that the path length is 31 and the number of red points is 3. From the result, we can observe that MMEA-SES can maintain all the optimal solutions even they have the same objective values. For traditional MOEAs, it's easy to find the Pareto front. However, finding all the equivalent solutions is a tough task. Due to the length limitation, other results can be found in https://github.com/Wenhua-Li/MMEASES and will not be described in this paper.

4.4. Embed special environmental selection into other MOEAs

To further study the effect of the proposed special environmental selection, we embed the special environmental selection method into NSGA-II, resulting in NSGA-II-SES. Specifically, for the original NSGA-II, the environmental selection is based on the fast non-dominated sorting and the crowding distance in the objective. Then, we embed the special environmental selection method into the beginning of the environmental selection of NSGA-II and remain other parts unchanged. Specifically, before using the original NSGA-II environmental selection, we first utilize the special environmental selection method proposed in Algorithm 2. Then, the mating selection of NSGA-II-SES is according to the Pareto front number and the crowding distance of the solutions.

Other parameters are set according to the experiments in Section 4.1 and all experiments are executed 31 times and the average result is shown in Table 4. As we can see, the performance of NSGA-II-SES on these problems is significantly better than that of NSGA-II. Notably, for problems 6 and 7, NSGA-II-SES can stably obtain all different optimal solutions. In addition, for other problems, the performance of NSGA-II-SES is only a little bit worse than MMEA-SES. Further analysis shows that by utilizing the special environmental selection, solutions in population during the evolution will not be re-duplicative and thus the diversity of the population can be well maintained. However, the performance of NSGA-II-SES is still worse than that of MMEA-SES. The reason is that the use of crowding distance in the objective space may force the algorithm to delete the potential optimal solutions. Moreover, in MMEA-SES, we introduce a second-selection method to ulteriorly improve the diversity.

5. Conclusion

Path planning problems are real-world problems that aim to find the optimal paths. In order to provide more information to the decision-makers and give them alternative solutions, we need to maintain the solution diversity in the decision space as well as keep the convergence quality in the objective space. Pitifully, few works focus on multi-modal discrete problems. In this paper, we proposed a novel evolutionary algorithm to solve this kind of problem. Specifically, in each generation of the algorithm, a special environmental selection strategy is performed to select offspring and thus maintain diversity. Through the experiments, the effectiveness and efficiency of our proposed algorithm have been proved.

Multi-modal multi-objective problems (MMOPs) are well-studied and many algorithms have been proposed. However, most of them are designed for continuous problems. The evaluation method of diversity in the decision space is highly related to the distance between solutions. For continuous problems, the distance is easy to calculate. However, the calculation method of solutions distance for discrete problems varies for different kinds of problems. We proposed another way to maintain the diversity and we don't need to evaluate the distance. This may be a feasible strategy for solving discrete MMOPs. In the future, more real-world engineering problems will be considered.

Acknowledgment

This work was supported by the National Science Fund for Outstanding Young Scholars (62122093), the Scientific key Research Project of National University of Defense Technology (ZZKY-ZX-11-04), the Ji-Hua Laboratory Scienctific Project (X210101UZ210) and the Special zone for Innovation of National Defense Technology (A11-203-01-01).

References

Coello, C. A. C., & Sierra, M. R. (2004). A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Mexican International Conference on Artificial Intelligence* (pp. 688–697). Springer.

Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems, 9*(2), 115–148.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*(2), 182–197.

Deb, K., & Tiwari, S. (2005). Omni-optimizer: A procedure for single and multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 47–61). Springer.

Deb, K., & Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research, 185*(3), 1062–1087.

Huang, G., Yuan, X., Shi, K., Liu, Z., & Wu, X. (2021). A 3-d multi-object path planning method for electric vehicle considering the energy consumption and distance. *IEEE Transactions on Intelligent Transportation Systems, PP*(99), 1–13.

Hu, C., & Ishibuchi, H. (2018). Incorporation of a decision space diversity maintenance mechanism into moea/d for multi-modal multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 1898–1901).

Ishibuchi, H., Imada, R., Setoguchi, Y., & Nojima, Y. (2018). Reference point specification in inverted generational distance for triangular linear pareto front. *IEEE Transactions on Evolutionary Computation, 22*(6), 961–975.

Jin, B. (2021). Multi-objective a* algorithm for the multimodal multi-objective path planning optimization. In *2021 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1704–1711). IEEE.

Kudo, F., Yoshikawa, T., & Furuhashi, T. (2011). A study on analysis of design variables in pareto solutions for conceptual design optimization problem of hybrid rocket engine. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 2558–2562). IEEE.

Liang, J., Yue, C., Li, G., Qu, B., Suganthan, P., & Yu, K. (2020). Problem definitions and evaluation criteria for the cec 2021 on multimodal multiobjective path planning optimization. Tech. Rep. Zhengzhou University and Nanyang Technological University.

Liang, J., Guo, Q., Yue, C., Qu, B., & Yu, K. (2018). A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In *International Conference on Swarm Intelligence* (pp. 550–560). Springer.

Liang, J., Xu, W., Yue, C., Yu, K., Song, H., Crisalle, O. D., et al. (2019). Multimodal multiobjective optimization with differential evolution. *Swarm and Evolutionary Computation, 44*, 1028–1059.

Liang, J., Yue, C., & Qu, B. (2016). Multimodal multi-objective optimization: A preliminary study. In *2016 IEEE Congress on Evolutionary Computation (CEC)* (pp. 2454–2461). IEEE.

Liu, Y., Ishibuchi, H., Nojima, Y., Masuyama, N., & Shang, K. (2018). A double-niched evolutionary algorithm and its behavior on polygon-based problems. In *International Conference on Parallel Problem Solving from Nature* (pp. 262–273).

Liu, Y., Ishibuchi, H., Yen, G. G., Nojima, Y., & Masuyama, N. (2019). Handling imbalance between convergence and diversity in the decision space in evolutionary multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 24*(3), 551–565.

Li, K., Wang, R., Zhang, T., & Ishibuchi, H. (2018). Evolutionary many-objective optimization: A comparative study of the state-of-the-art. *IEEE Access, 6*, 26194–26214.

Li, W., Wang, R., Zhang, T., Ming, M., & Lei, H. (2019). Multi-scenario microgrid optimization using an evolutionary multi-objective algorithm. *Swarm and Evolutionary Computation, 50*, 100570.

Li, W., Wang, R., Zhang, T., Ming, M., & Li, K. (2020). Reinvestigation of evolutionary many-objective optimization: Focus on the pareto knee front. *Information Sciences, 522*, 193–213.

Li, W., Yao, X., Zhang, T., Wang, R., & Wang, L. (2022). Hierarchy ranking method for multimodal multi-objective optimization with local pareto fronts. *IEEE Transactions on Evolutionary Computation*. Early Access:1–1.

Li, W., Zhang, T., Wang, R., & Ishibuchi, H. (2021). Weighted indicator-based evolutionary algorithm for multimodal multiobjective optimization. *IEEE Transactions on Evolutionary Computation, 25*(6), 1064–1078.

Li, W., Zhang, G., Yang, X., Tao, Z., & Xu, H. (2021). Sizing a hybrid renewable energy system by a coevolutionary multiobjective optimization algorithm. *Complexity, 2021*.

Li, W., Zhang, G., Zhang, T., & Huang, S. (2020). Knee point-guided multiobjective optimization algorithm for microgrid dynamic energy management. *Complexity, 2020*.

Rudolph, G., Naujoks, B., & Preuss, M. (2007). Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 36–50). Springer.

Rudolph, G., & Preuss, M. (2009). A multiobjective approach for finding equivalent inverse images of pareto-optimal objective vectors. In *2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM)* (pp. . 74–7). IEEE.

Schütze, O., Vasile, M., & Coello Coello, C. A. (2011). Computing the set of epsilon-efficient solutions in multiobjective space mission design. *Journal of Aerospace Computing, Information, and Communication, 8*(3), 53–70.

Sebag, M., Tarrisson, N., Teytaud, O., Lefevre, J., & Baillet, S. (2005). A multi-objective multi-modal optimization approach for mining stable spatio-temporal patterns. In *IJCAI* (pp. 859–864).

Siddiqi, M. R., Milani, S., Jazar, R. N., & Marzbani, H. (2021). Ergonomic path planning for autonomous vehicles-an investigation on the effect of transition curves on motion sickness. *IEEE Transactions on Intelligent Transportation Systems, PP*(99), 1–12.

Tanabe, R., & Ishibuchi, H. (2018). A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature* (pp. 249–261). Springer.

Tanabe, R., & Ishibuchi, H. (2019). A niching indicator-based multi-modal many-objective optimizer. *Swarm and Evolutionary Computation, 49*, 134–146.

Wang, R., Ishibuchi, H., Zhou, Z., Liao, T., & Zhang, T. (2018). Localized weighted sum method for many-objective optimization. *IEEE Transactions on Evolutionary Computation, 22*(1), 3–18.

Yue, C., Qu, B., & Liang, J. (2017). A multiobjective particle swarm optimizer using ring topology for solving multimodal multiobjective problems. *IEEE Transactions on Evolutionary Computation, 22*(5), 805–817.

Zhao, J., Jia, Z., Zhou, Y., Zhang, R., Xie, Z., Xu, Z., et al. (2021). Path planning based on multi-objective topological map. In *2021 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1719–1726). IEEE.

Zitzler, E., & Künzli, S. (2004). Indicator-based selection in multiobjective search. In *International Conference on Parallel Problem Solving from Nature* (pp. 832–842). Springer.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation, 7*(2), 117–132.