2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

Improved Evolutionary Strategies for Sparse Large-Scale Many-objective Optimization Problems

Jiawei Chen[1#]
College of Systems Engineering
National University of Defense Technology
Changsha, China
Email: 857248498@qq.com

Lei He[#*]
College of Systems Engineering
National University of Defense Technology
Changsha, China
Email: helei@nudt.edu.cn

Yingwu Chen
College of Systems Engineering
National University of Defense Technology
Changsha, China
Email: ywchen@nudt.edu.cn

Abstract—**Multi-objective optimization problems with various attributes are studied for two decades. Sparsity, as one of them, sparked many researchers. However, they usually focused on sparse large-scale bi-objective optimization. The result is unsatisfying when applying their algorithms to optimization problems with more than three objectives. To solve this issue, this paper selects a classical algorithm for large-scale sparse multi-objective optimization problems and proposes the reference points and adaptive crossover and mutation strategies to the original algorithm, adapting it to the sparse many-objective optimization problem. After a series of experiments, the algorithm with this modification mostly dominates other state-of-the-art multi-objective optimization algorithms. Although several best performance metrics are obtained from other competitors, the highest sparsity on the Pareto optimal solution is still completed by the proposed algorithm.**

Keywords—Evolutionary algorithm, large-scale multi-objective optimization (MOP), sparse Pareto optimal solutions, many-objective optimization.

I. INTRODUCTION

Multi-objective optimization problems (MOPs) refer to problems with contradictory and mutually influenced objectives and the problem is solved when its objectives are optimized simultaneously, which are studied in various engineering fields for decades. Back in 1998, Schaumane[1] utilized Genetic Algorithm (GA) for the MOP of a reinforced concrete structure and for an urban planning problem. Concrete Problem is to minimize material cost and construction time while the urban problem aims to minimize the traffic travel time, the cost, and the change in land use. For now, Wei[2] design a Memetic Approach to solve the time-dependent agile earth observation satellite scheduling problem with two objectives: maximizing the weights of completed tasks and the imaging quality of the observation task. In a MOP, because its objectives are conflict with each other, there is no perfect solution with the best optimization for every objective. In order to provide adequately satisfactory solutions, Pareto optimal solution is proposed. It includes a set of solutions with a common nature that there does not exist a solution that is better in one objective and not worse in other objectives. Thus, finding the Pareto optimal solutions is the goal of solving MOPs.

The multi-objective evolutionary algorithm (MOEA) is a kind of intelligent algorithm, aiming to solve MOPs. Compared to the mathematical programming methods, which can quickly obtain a single solution in high-dimensional space, the MOEA is an expert in the tradeoff among conflicting objectives and generates a set of Pareto optimal solutions. However, due to the exponentially increasing decision space with the number of decision variables, traditional MOEA coverage is very slow, known as the *curse of dimensionality*[3]. To improve them, many operators and search strategies are proposed to solve Large-Scale Multi-Objective Optimization Problems (LSMOPs)[4, 5] according to the specific attribute of problems, such as constrained[6], expensive[7], multimodal[8], sparse[9], preference[10] and so on.

Among these attributes, the sparse LSMOP means the most decision variables of the Pareto optimal solutions are zero. There are many applications for sparse LSMOP, such as feature selection[11], critical node detection[12], neural network training[13], pattern mining[14], and so on. In order to solve this kind of problem, Tian proposed an MOEA for solving sparse LSMOPs, called SparseEA[9]. This algorithm contains a similar framework with NSGA-II in mating and environmental selection since they both calculate and utilize the non-dominated front number and crowding distance. The creativity of Tian's research is the masking strategy to guarantee the sparsity of the initial population. Based on it, many studies were made for improving its effectiveness. Kropp[15] proposed three evolutionary operators: varied striped sparse population sampling (VSSPS), sparse simulated binary crossover (S-SBX), and sparse polynomial mutation (S-PM), combined with NSGA-II, to greatly improve the algorithm performance. Jing[16] introduces a two-stages strategy which is decreasing dimensionality with low-dimensional binary weights and facilitating offspring evolution by the new hybrid encoding. Zou[17] proposed a dynamic sparse grouping strategy to group decision variable with a comparable amount of nonzero values and optimize them in one group, which achieve better performance than SparseEA in benchmark problems.

However, these researchers all improve their algorithm in the same experiment scenario and the benchmark problems are all related to two objectives. In fact, the sparse LSMOPs

[1#] The authors contributed equally to the paper. *Corresponding author highly possibly contain more than two objectives, which are called many-objectives. For example, a neural network training problem, which is verified as a sparse LSMOP, may contain many objectives, such as optimizing the topology, weight, hyperparameters, diversity, and accuracy of the network[18]. In addition, Zhang[14] set the patterning mining, a sparse LSMOP, as a bi-objective maximum problem and discover that two objectives cannot distinguish the importance of patterns in some contexts. In his research, the objectives in terms of support and occupancy cannot determine which is better between the two patterns because items in real life possess numerous features. Therefore, MOEA should be further studied for solving sparse LSMOP with many objectives

We applied PlatEMO[19], a Matlab platform for evolutionary multi-objective optimization, to test the effectiveness of SparseEA for many-objective optimization problems.

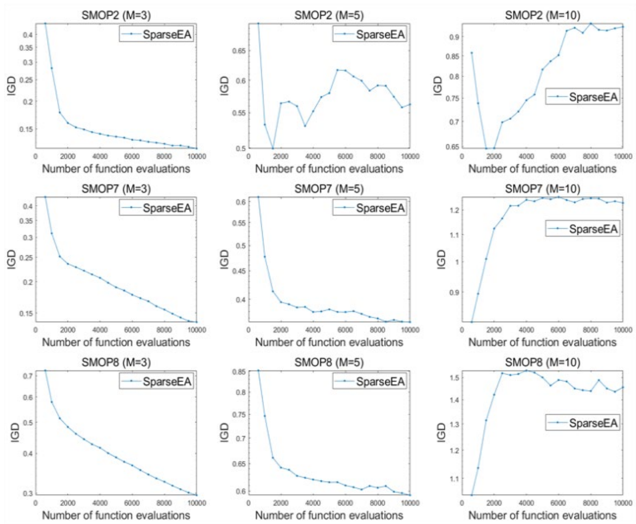


Fig. 1. IGD Metric in Evolutionary Computation of SparseEA on Different Benchmark Problems with Many-Objectives

Fig. 1 displays the Inverted Generational Distance (IGD)[20] index of SparseEA for different many-objective problems with the increasing number of function evaluations. In Fig. 1, SMOP2, SMOP7, and SMOP8 are different benchmark problems and M is the number of objectives. Additionally, since IGD is the distance between obtained solution and the known optimal solution, the smaller value of IGD indicates the better performance of the MOEA. And with the increasing of number of evaluations, the IGD should decrease with an efficient algorithm, implying that evolutionary computing works. However, the IGD index is even increasing in some cases and exponentially increasing with more objectives, so SparseEA cannot solve many-objectives problems effectively. To solve this issue, this paper proposes an improved SparseEA with Adaptive Cross-Mutation (ACM) mechanism and Reference Point (RP) to solve the many-objective sparse large-scale optimization problem (SparseEA-M).

The main contributions made by this paper are summarized as follows:

(1) This paper firstly proposes the sparse many-objective problems with high value of research and some targeted strategies on the original SparseEA so that it is capable of solving many-objective problems.

(2) We verify the RP, from NSGA-III, still performs well on sparse many-objective problems. Additionally, since the original evolutionary variant only contains bare crossover and mutation strategy, modification of this process is necessary and ACM, proposed in this paper, is an efficient way of improving the algorithm performance.

(3) Our experiments indicate that the proposed algorithm dominates on most problems and although some problems do not, the higher sparsity can be still confirmed.

The rest of this paper is organized as follows. Section II introduces the original structure of SparseEA, the modifications proposed in this paper, and the whole structure of SparseEA-M. Section III describes the experiment settings and results. Lastly, the conclusion is made in Section IV.

II. Evolutionary Algorithm for Sparse Many-Objective Optimization Problems

A. The Framework of the Original SparseEA

The structure of the original SparseEA contains two main steps: 1) Population initialization: firstly calculating the score of every individual decision variable by the Pareto non-dominated front number and crowding distance. And randomly comparing two decision variables and setting its corresponding mask variable to 1 as the winner while that of the loser is 0, which should be multiplied by the real decision variables. In this way, the significant decision variables with higher scores are counted by multiplying with 1 while those peripheral parts are ignored by multiplying with 0 for generating the initial population, as the indispensable way of guaranteeing the sparsity; 2) Evolutionary variation: randomly selecting two parents and applying crossover and mutation strategies to change the values of the real decision variables and binary mask variables. Specifically, the crossover strategy is to randomly select and compare two non-zero elements from two parents' mask variables according to their scores, and then, with each 1/2 probability, set the corresponding offspring's mask variable of the winner to 1 or the loser to 0, ensuring the optimal of solutions. The mutation strategy is to randomly select and compare two non-zero elements from the offspring's mask variables according to their scores, and then, with each 1/2 probability, set the corresponding offspring's mask variable of the winner to 0 or the loser to 1, ensuring the diversity of solutions.

However, the primary framework of the SparseEA possesses some disadvantages. The non-dominated front number and crowding distance are naturally unsuitable for analyzing many objectives, because the non-dominated front number exponentially increases with higher dimension of objectives, causing inefficient comparisons among solutions, and calculating crowding distance among many objectives is arduous. Furthermore, the many-objective property also renders some challenges such as difficult convergence and solution distribution on Pareto Fronts (PFs). In order to ameliorate the original algorithm, inspired by NSGA-III[21], this paper introduces the RP to SparseEA. The purpose of the RP is to obtain the mapping relationship, the vertical distance, between the individual population and the corresponding RP, so that the population evolves closer to the RPs and the distribution of the RP is more uniform. Secondly, after the scoring of population initialization, the scores are not changed and without a relatively directed mechanism helping evolutionary variation, causing the crossover and mutation to be disorganized and wasteful. For this issue, this paper proposed ACM) to improve the effectiveness of the SparseEA.

B. The Framework of the Proposed SparseEA-M

The original SparseEA is divided into two parts: population initialization and evolutionary variant. Since the binary mask variables and scoring of every decision variable are creative and efficient, this paper adds improvements mainly on the second part. Fig. 2 shows the whole algorithm flowchart with two process parts and green highlighted parts are most of the proposed modifications. From those green highlighted parts, the first changed part is that parents are generated via binary tournament selection according to the non-dominated number and RP distances of solutions, instead of crowding distances from the previous algorithm. The second improved part is the PRs and PR distances should be updated for one evolutionary loop. The adapted parts from 3 to 5 are related to the ACM mechanism. Based on the original crossover and mutation strategy with fixed probability, adaptive changes according to the Pareto rank of solutions are proposed. In this way, the better solutions should be selected for evolution with less probability since they may require less improvement while the worse solutions should be selected with a related higher probability for optimizing the whole population.

For the two modifications, Firstly, RP refers to the selection of some representative target points in the multi-objective optimization problem to evaluate the advantages and disadvantages of each solution. These target points should cover the entire PF as much as possible so that more un-dominated solutions can be found. The utilization of RP can effectively reduce the search space and improve search efficiency. Secondly, Adaptive Crossover and Mutation refers to the strategy of adaptively adjusting the crossover and mutation according to the state of the current population in a many-objective optimization problem. This allows the algorithm to better adapt to the current search space and avoid falling into the local optimal solution. The adaption of ACM can improve the convergence speed and search efficiency of the algorithm. Therefore, they both play significant roles in many-objective optimization problems, which can help the algorithm better adapt to the characteristics of the problem and improve search efficiency and accuracy.

For creating RP, firstly, define *M-dimensional* RPs set $V = \{v_1, v_2, \dots, v_M\}$. One coordinate value of RPs is $v_j \in \{\frac{0}{d}, \frac{1}{d}, \dots, \frac{d}{d}\}$ where $\sum_{j=1}^{M} v_j = 1$ and d is the number of divisions for every objective. Let B be (*M-1*) combinations of $\{\frac{0}{d}, \frac{1}{d}, \dots, \frac{d+M-2}{d}\}$, and change each $b_{ij} \in B$ to $b_{jl} - \frac{j-1}{d}$, where j is the index of element and l is the index of combination. Lastly, the coordinate value of RPs is calculated:

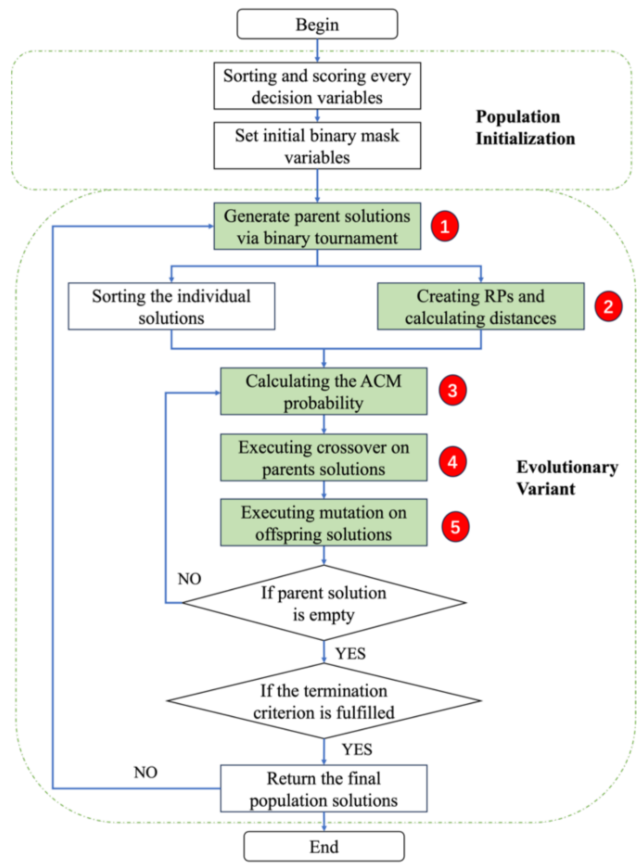


Fig. 2. The proposed SparseEA-M algorithm framework. The process is divided into two parts: Population initialization and Evolutionary Variant. The green highlighted parts are the improvements. The red nodes label the improved parts.

$$\begin{cases} v_{jl} = b_{ij} - 0,\ j = 1 \\ v_{jl} = b_{jl} - b_{(j-1)l},\ \ 1 < M < j \\ v_{jl} = 1 - r_{(j-1)l},\ j = M \end{cases} \tag{1}$$

According to the RPs' coordinate values from Equation (1), the solutions calculated during evolution would be get closer to these RPs. Specifically, the distance between the individual and RPs influences this solution's Pareto rank. Less distance means the optimal of the individual. This improvement promotes the optimal solution distribution on PFs.

ACM mechanism provides the adaptive crossover probability P_c and mutation probability P_m in evolutionary variation. The crossover strategy uses the method of partially matching crosses. The ACM mechanism is utilized to increase the evolutionary opportunities of superior individuals and improve the searchability of the algorithm. The adaptive crossing probability is:

$$P_{ci} = P_c * \frac{\max(r_1, r_2, \ldots r_n) - r_i + 1}{\max(r_1, r_2, \ldots r_n)} \quad (2)$$

$$P_{mi} = P_m * \frac{\max(r_1, r_2, \ldots r_n) - r_i + 1}{\max(r_1, r_2, \ldots r_n)} \quad (3)$$

where P_{ci} is the adaptive crossover probability of individual i, P_c is the pre-set probability, n is the population size, and r_i is the Pareto rank of individual i, and where P_{mi} is the adaptive mutation probability of individual i and P_m is the pre-set probability

According to Equations (2) and (3), the crossover and mutation probability are both changed with the Pareto rank of individuals, which is obtained by the non-dominated front number. A higher Pareto rank renders lower crossover and mutation probability for every individual.

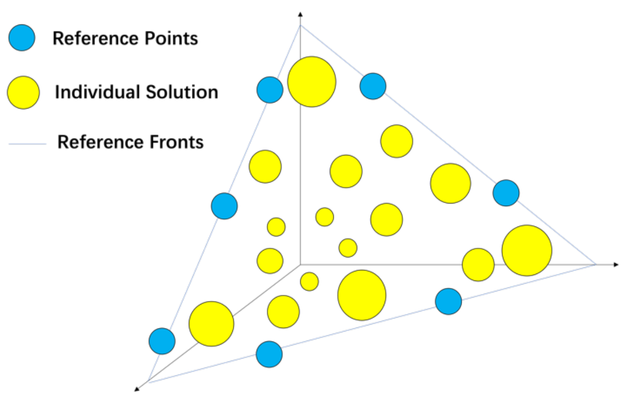


Fig. 3. Reference Points on Pareto Fronts and Solution Population (M = 3)

Fig. 3 displays the relationship between RPs and the solution population. Since there are three objectives in this case, this figure is presented in three dimensions. In Fig. 2, the size of an individual indicates the Pareto rank. The larger nodes imply that it has a higher Pareto rank. According to Equations (2) and (3), these individuals on the higher Pareto rank would possess a lower probability to be selected for crossover and mutation, since they are closer to the optimal solutions, which accelerates coverage and maintains the optimal.

Algorithm 1 is the pseudocode of the framework of the proposed SparseEA-M. Lines 1-6 complete the population initialization, containing the initial mask variable and decision variable, Pareto rank, RPs, and their distances. Lines 7-21 accomplish the evolutionary variations. Within this process, except for the original strategy, the ACM and RPs are added. The crossover and mutation strategies are made on lines 9-15. In these two processes, P_c is the crossover probability of selecting two individual parents, while P_m is the mutation probability of selecting two decision variables. After this variation, lines 16-21 update the Pareto rank, RPs and distances, and the population for the next evolution. Until the termination criterion is fulfilled, the final population, the Pareto optimal solution, is obtained.

III. Experimental Results

For confirming the beneficial effects from the improvements, we first compare IGD in evolutionary computing by the original SparseEA and the proposed SparseEA-M. And then we test the original SparseEA, SparseEA-M, NSGA-III[21], MOEA/D-DRA, and CMOPSO on eight benchmark problems. The first two algorithms are introduced in Section II. NSGA-III, the first algorithm introducing RPs, is the upgrade on NSGA-II. MOEA/D is the algorithm separating many objectives into individual single objective with different weights and applying a weighted sum approach to integrate all objectives. MOEA/D-DRA introduces the Dynamic Resource Allocation (DRA) strategy to the original MOEA/D for extremely decreasing computing time, making it possible to solve many-objective problems. CMOPSO[22] proposes a

Algorithm 1 Framework of the Proposed SparseEA-M

Input: N (Population Size)

Output: PO (Final Population)

```
1  D ← Number of decision variables;
2  M ← Number of objectives;
3  O ← ∅
4  [PO, Score] ← Initialization(N); // Calculate the score of
   every individual decision variable and set the mask
   variable according to the original SparseEA
5  [F1, F2, F3, ...] ← Do non-dominated sorting on PO;
6  V ← ReferenceePointsDistance(N,M,F); // Creating PRs and
   calculate the distances between solution F and PRs.
7  While Termination criterion not fulfilled do
8      PO' ← Select 2N parents via binary tournament selection
       according to the non-dominated front number and Z;
9      While PO' is not empty do
10         Pc, Pm ← calculate crossover and mutation probability;
11         [p, q] ← select two parents based on crossover probability
           Ppc and Pqc ;
12         [o.mask] ← Crossover(p, q, Score); // The crossover
           strategy from the original SparseEA.
13         [x1, x2] ← select two nonzero variable in o.mask based on
           mutation probabilities Px1m and Px2m;
14         [o.mask] ← Mutation( x1 , x2 , o.mask, Score); // The
           mutation strategy from the original SparseEA.
15         [o.dec] ← GenerateDec(o.dec, p.dec, q.dec); // Update the
           real decision variables in the offspring based
           on original SparseEA.
16     Delete duplicated solutions from PO;
17     [F1, F2, F3, ...] ← Do non-dominated sorting on PO;
18     V ← ReferenceePointsDistance(N,M,F); // Update PRs and
       calculate the distances between solution F and PRs.
19     k ← argmin_i | F1 ∪ ... ∪ Fn | >= Ni
20     Delete | F1 ∪ ... ∪ Fn | − N solutions from Fk with the
       largest RPs distance;
21     PO ← F1 ∪ ... ∪ Fn ;
22 Return PO;
```

competitive mechanism to particle swarm optimization (PSO) for extending the original PSO to solve multi-objective optimization. These algorithms are the state-of-the-art algorithms for solving sparse multi-objective problems.

The eight benchmark problems SMOP1 to SMOP8, with known Pareto optimal solutions, are proposed by Tian[9]. Tian analyzed the existing multi-objective benchmark problems (MOPs) and found their decision variables are not sparse on Pareto optimal solutions. To make them sparse, in short, the crucial modified part is to multiply a ratio θ ($\theta \in [0,1]$) with a parameter K, which indicates the number of decision variables related to the two landscape functions. Because this modification decreases K and a Pareto set of one example is $[0,1]^{M-1} \times \{\pi/3\}^{K} \times \{0\}^{D-M+1-K}$ from their research, when K is smaller, the number of zero decision variables is increasing and the sparsity is achieved. More information refers to Tian's study.

A. Parameters Setting

1) Problems: For SMOP1 to SMOP8, we set the population size N as 100, the numbers of objectives are 3, 5, and 10, the number of decision variables is 500, and the ratio θ is 0.1 since the smaller value of θ implies a higher sparsity.

2) Operators: the proposed basic crossover and mutation probability P_c and P_m are 1 and $1/D$, respectively. Similarly, the crossover and mutation probability of NSGA-III and SparseEA are 1 and $1/D$. Additionally, in differential evolution in MOEA/D-DRA, the parameter CR is 1 and F is 0.5.

3) Termination criterion: the number of function evaluations for each test of one algorithm and one benchmark problem is $100 \times D$.

4) Performance Metric: The inverted generational distance (IGD)[20] is an index to measure the distances between obtained solutions and Pareto optimal solutions. The smaller value of IGD indicates better performance. Approximately 1000 optimal solutions on each PF of SMOP1 to SMOP8 are sampled by the proposed methods[9]. Each benchmark problem is run 30 times for abundant data results. In addition, the Wilcoxon rank sum test with a significance level of 0.05 and the Holm procedure[23] is made. The symbol "+," "-," "≈" means that the result from this algorithm is better, much worse, and similar to that of the proposed algorithm.

5) Programming Environment: All experiments all made in the PlatEMO platform under MATLAB 2022a.

TABLE I. MEDIAN IGD VALUES AND IQRS FROM NSGA-III, CMOPSO, MOEA/D-DRA, SPARSEEA, AND SPARSE-M ON SMOP1 TO SMOP8, WHERE THE BEST RESULT OF EVERY ROW IS HIGHLIGHTED

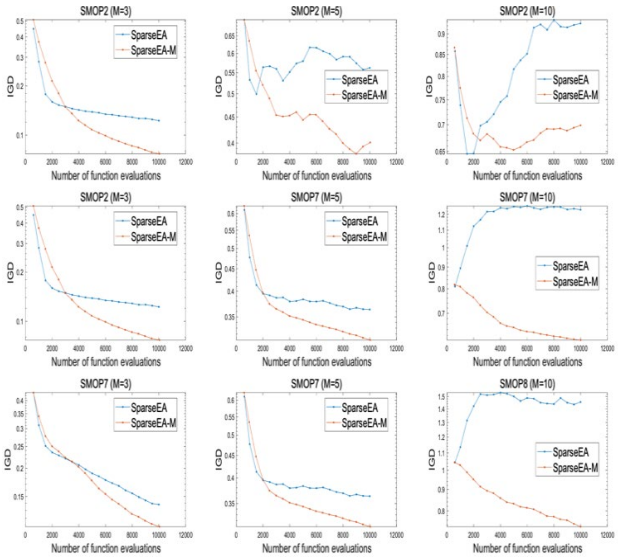
Problems	M	D	NSGA-III	CMOPSO	MOEA/D-DRA	SparseEA	SparseEA-M
	3	500	2.2204e-1 (6.21e-3) -	7.3664e-1 (3.88e-2) -	6.2285e-1 (2.97e-2) -	7.7108e-2 (3.77e-3) -	**4.7936e-2 (1.78e-3)**
SMOP1	5	500	**4.1919e-1 (5.67e-3) +**	1.4826e+0 (7.31e-2) -	7.3120e-1 (4.94e-2) -	4.3987e-1 (2.72e-2) +	5.5286e-1 (1.87e-1)
	10	500	1.1278e+0 (2.12e-2) -	1.4722e+0 (1.21e-1) -	7.1919e-1 (4.81e-2) ≈	7.8077e-1 (6.41e-2) -	**6.7867e-1 (8.31e-2)**
	3	500	8.4262e-1 (1.80e-2) -	1.7389e+0 (2.65e-2) -	1.6843e+0 (1.51e-2) -	1.2679e-1 (7.20e-3) -	**7.9664e-2 (4.69e-3)**
SMOP2	5	500	1.0173e+0 (1.15e-2) -	1.5446e+0 (3.86e-2) -	1.4331e+0 (3.51e-2) -	6.0520e-1 (9.58e-2) ≈	**5.2584e-1 (1.62e-1)**
	10	500	1.8308e+0 (1.26e-2) -	1.4645e+0 (5.72e-2) -	1.1231e+0 (2.50e-2) -	9.2880e-1 (4.07e-2) -	**7.3493e-1 (5.83e-2)**
	3	500	9.1202e-1 (1.97e-2) -	1.6584e+0 (6.79e-2) -	1.4370e+0 (3.24e-2) -	**8.3991e-2 (3.73e-3) +**	1.3801e-1 (1.95e-1)
SMOP3	5	500	1.1395e+0 (2.00e-2) -	1.8182e+0 (4.28e-2) -	1.3137e+0 (4.76e-2) -	7.5168e-1 (1.97e-1) -	**5.5680e-1 (1.04e-1)**
	10	500	2.1766e+0 (1.52e-2) -	1.6354e+0 (6.02e-2) -	1.1126e+0 (2.91e-2) -	9.3613e-1 (1.31e-1) -	**6.4587e-1 (4.56e-2)**
	3	500	3.1424e-1 (5.72e-3) -	6.3304e-1 (1.04e-2) -	6.2858e-1 (6.33e-3) -	6.1171e-2 (8.52e-3) -	**2.6287e-2 (1.66e-4)**
SMOP4	5	500	2.8293e-1 (4.64e-3) ≈	4.5212e-1 (2.88e-2) ≈	3.6288e-1 (1.29e-2) ≈	**2.2479e-1 (2.95e-2) ≈**	5.4143e-1 (4.10e-1)
	10	500	3.1208e-1 (7.53e-2) +	4.7493e-1 (6.14e-2) +	**1.8885e-1 (3.23e-3) +**	3.6021e-1 (2.88e-2) +	7.0207e-1 (2.70e-1)
	3	500	2.4126e-1 (1.46e-3) -	4.0601e-1 (1.56e-2) -	2.7900e-1 (2.61e-3) -	6.4107e-2 (5.98e-3) -	**2.6610e-2 (1.62e-4)**
SMOP5	5	500	**2.0752e-1 (3.15e-3) +**	4.2608e-1 (4.70e-2) +	2.2949e-1 (1.19e-2) +	3.5238e-1 (2.50e-1) +	7.6169e-1 (1.38e-1)
	10	500	2.8149e-1 (6.42e-2) +	5.1023e-1 (8.91e-2) ≈	**1.6552e-1 (1.67e-2) +**	8.2332e-1 (1.24e-1) -	5.3081e-1 (1.05e-1)
	3	500	6.7640e-2 (1.24e-3) -	2.1428e-1 (7.66e-3) -	1.4741e-1 (4.11e-3) -	6.4813e-2 (6.32e-3) -	**2.6859e-2 (2.38e-4)**
SMOP6	5	500	**1.1893e-1 (1.79e-3) +**	1.9771e-1 (1.40e-2) +	1.8185e-1 (7.96e-3) +	4.0457e-1 (3.27e-1) +	6.6644e-1 (2.20e-1)
	10	500	1.8389e-1 (3.25e-2) +	2.7762e-1 (2.46e-2) +	**1.2568e-1 (2.95e-3) +**	8.2000e-1 (1.22e-1) -	5.7837e-1 (9.60e-2)
	3	500	5.6527e-1 (2.89e-2) -	8.6240e-1 (3.27e-2) -	8.5978e-1 (6.28e-2) -	1.7320e-1 (9.38e-3) -	**1.1438e-1 (9.68e-3)**
SMOP7	5	500	1.2535e+0 (4.58e-2) -	3.7296e+0 (5.34e-1) -	1.3956e+0 (1.19e-1) -	3.7311e-1 (1.69e-2) -	**2.9024e-1 (4.12e-3)**
	10	500	2.4263e+0 (5.90e-2) -	4.7569e+0 (9.57e-1) -	1.6982e+0 (9.92e-2) -	1.3125e+0 (7.42e-2) -	**5.7608e-1 (2.25e-2)**
	3	500	2.9989e+0 (1.26e-1) -	3.6143e+0 (2.42e-2) -	2.8004e+0 (2.59e-1) -	3.3027e-1 (2.10e-2) -	**2.7345e-1 (1.69e-2)**
SMOP8	5	500	3.4648e+0 (2.37e-2) -	3.7669e+0 (1.25e-2) -	3.3089e+0 (3.62e-1) -	5.9101e-1 (1.38e-2) -	**4.7682e-1 (1.42e-2)**
	10	500	3.8017e+0 (3.54e-2) -	3.8349e+0 (7.97e-3) -	3.8896e+0 (1.29e-1) -	1.5121e+0 (6.87e-2) -	**7.7258e-1 (6.03e-2)**
+/-/≈			6/17/1	4/18/2	5/17/2	5/17/2	

B. Experimental Results

For verifying the improvements of the proposed SparseEA-M, the same test on IGD metric in the evolutionary computation of SparseEA-M is made and the comparison chart is shown in Fig.4. From this figure, the red line is SparseEA-M, which distinctly indicates a lower value and gradually decreases with the number of function evaluations. Since lower values mean better performance, the improvement is confirmed.

With these parameter settings, these five selected algorithms are tested and the results are shown in Table I, which contains the median IGD values and interquartile ranges. Obviously, the most of best results are on SparseEA-M, indicating the modifications are effective although there are several best results on other algorithms. The reason why some best results are not on SparseEA-M is that other algorithms are naturally experts in many-objective optimization. However, this phenomenon cannot deny the contribution of the improvements since the SparseEA-M still dominated most of the best results and it succeed in higher sparsity.

Since the PFs of an optimization problem with more than three objectives are not visualized distinctly in imaging, this



paper displays the Pareto optimal sets with IGD medians. Unexpectedly, the decision variable distribution indicates that the SparseEA-M shows the highest sparsity even though there are no specific modifications for increasing sparsity. To reveal this point, we choose three experimental examples. Normally,

Fig. 4 IGD Metric in Evolutionary Computation of SparseEA and SparseEA-M on Different Benchmark Problems with Many-Objectives

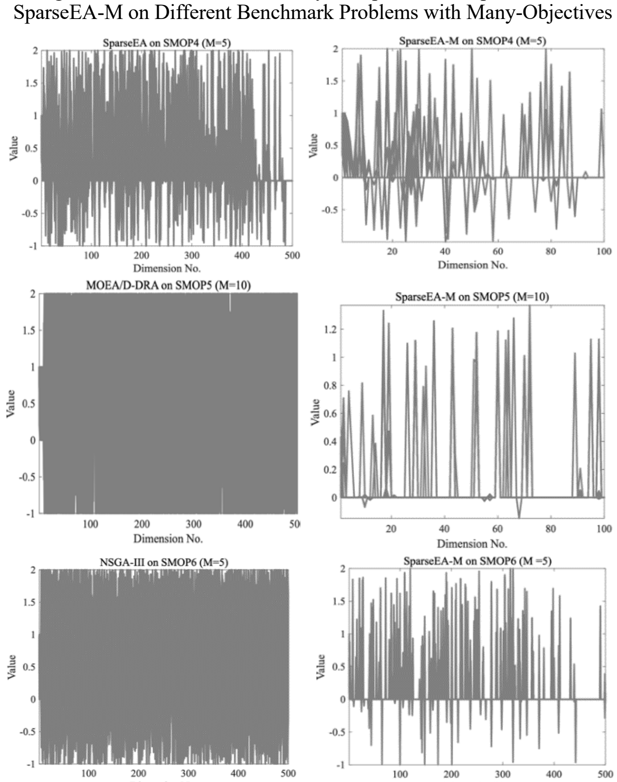


Fig. 5 SparseEA and the best competitive algorithms on SMOP4 to SMOP6 on different M with 500 decision variables over 30 independent runs produced Pareto optimal sets with median IGD value

according to the designing theory of benchmark problems SMOP1 to SMOP8, the higher sparsity highly suggests better final solutions. For those problems that the SparseEA-M does not dominate other competitors, the algorithms with the best result should highly possibly have higher sparse decision variables of their Pareto sets. However, the experimental results do not support this view. Fig.5 displays the distribution of all decision variables affecting median IGD, implying the number of nonzero decision variables and sparsity on Pareto set. In addition, these three pairs of graphs all demonstrate the comparison between the algorithm with the best IGD values and the SparseEA-M in the same experimental scenario. For example, Fig.4 compares the sparsity of the MOEA/D-DRA, which completes the best IGD values on SMOP5 (M=10), and that of the SparseEA-M. All figures prove that the SparseEA-M achieves higher sparsity than other algorithms although the competitors succeed in the best performance metrics. Therefore, the SparseEA-M unfolds the strong ability to complete the Pareto set with sparse decision variables. But the reason why it cannot achieve the best performance metric is that although sparsity is guaranteed, in other words, the zero binary mask variables are discovered previously, and the values of the non-zero decision variables are not optimized adequately. In future work, how to optimize the real decision variables is meaningful and significant research.

IV. Conclusion

Among MOPs, sparsity in decision variables is the one of important attributes. There are many real-world applications with sparse MOPs, such as patterning. By proposing some targeted strategies, the algorithm would perform with fast convergence and efficient optimizations. Recently, although many studies are made for this kind of problem, the problems they focus on are usually bi-objectives. Those problems with more than three objectives are called many-objective, which is also an important attribute. However, studies made on this are absent. To fill this gap, this paper introduces the Reference Point (RP) and (Adaptive Crossover and Mutation) ACM mechanism into the original SparseEA, which is verified as a successful algorithm for solving sparse MOPs. We set a series of experiments for testing our algorithm SparseEA-M, competing with other state-of-the-art multi-objective optimization algorithms. The experiment results indicate that SparseEA-M dominates other algorithms in most instances, although there are several best results obtained from other algorithms. Because SparseEA-M only dominates one test on SMOP4, we take it as an example to show its decision variable distribution. From this experiment result, SparseEA-M still achieves the highest sparsity. Therefore, these modifications certainly improve the performance of the original SparseEA.

To our best knowledge, this paper firstly studies the sparse many-objective optimization problems. Furthermore, we find the RP also works well on sparse many-objective optimization and efficient modification on evolutionary variant is necessary. In addition, the proposed algorithm achieves higher sparsity although it does not dominate other algorithms in performance metrics.

In future work, other adapted strategies would be tested to further improve the algorithm for solving many-objective optimization. In addition, other innovative strategies or algorithms for increasing sparsity should also be studied.

Acknowledgements: This work is supported by the National Natural Science Foundation of China (72001212) and Young Elite Scientists Sponsorship Program by CAST (2022QNRC001).

REFERENCES

[1] Schaumann, E., R. Balling, and K. Day. *Genetic Algorithms with multiple objectives*.

[2] Wei, L., et al., *A Multi-objective Memetic Approach for Time-dependent Agile Earth Observation Satellite Scheduling Problem.* Computers & Industrial Engineering, 2021. **159**.

[3] Tian, Y., et al., *Solving Large-Scale Multiobjective Optimization Problems With Sparse Optimal Solutions via Unsupervised Neural Networks.* IEEE Trans Cybern, 2021. **51**(6): p. 3115-3128.

[4] Qi, S., et al., *A level-based multi-strategy learning swarm optimizer for large-Scale multi-objective optimization.* Swarm and Evolutionary Computation, 2022. **73**.

[5] Qi, S., et al., *A self-exploratory competitive swarm optimization algorithm for large-scale multiobjective optimization.* Information Sciences, 2022. **609**: p. 1601-1620.

[6] Ming, F., et al., *A Competitive and Cooperative Swarm Optimizer for Constrained Multi-objective Optimization Problems.* IEEE Transactions on Evolutionary Computation, 2022: p. 1-1.

[7] Wang, H., et al., *An adaptive batch Bayesian optimization approach for expensive multi-objective problems.* Information Sciences, 2022. **611**: p. 446-463.

[8] Li, W., et al., *Multimodal multi-objective optimization: Comparative study of the state-of-the-art.* Swarm and Evolutionary Computation, 2023. **77**.

[9] Tian, Y., et al., *An Evolutionary Algorithm for Large-Scale Sparse Multiobjective Optimization Problems.* IEEE Transactions on Evolutionary Computation, 2020. **24**(2): p. 380-393.

[10] Dukovska, I., J.G. Slootweg, and N.G. Paterakis, *Introducing user preferences for peer-to-peer electricity trading through stochastic multi-objective optimization.* Applied Energy, 2023. **338**.

[11] Xue, B., M. Zhang, and W.N. Browne, *Particle swarm optimization for feature selection in classification: a multi-objective approach.* IEEE Trans Cybern, 2013. **43**(6): p. 1656-71.

[12] Lalou, M., M.A. Tahraoui, and H. Kheddouci, *The Critical Node Detection Problem in networks: A survey.* Computer Science Review, 2018. **28**: p. 92-117.

[13] Zhang, C., et al., *Multiobjective Deep Belief Networks Ensemble for Remaining Useful Life Estimation in Prognostics.* IEEE Trans Neural Netw Learn Syst, 2017. **28**(10): p. 2306-2318.

[14] Zhang, X., et al., *Pattern Recommendation in Task-oriented Applications: A Multi-Objective Perspective [Application Notes].* IEEE Computational Intelligence Magazine, 2017. **12**(3): p. 43-53.

[15] Kropp, I., A.P. Nejadhashemi, and K. Deb, *Improved Evolutionary Operators for Sparse Large-Scale Multiobjective Optimization Problems.* IEEE Transactions on Evolutionary Computation, 2023: p. 1-1.

[16] Jiang, J., et al., *A two-stage evolutionary algorithm for large-scale sparse multiobjective optimization problems.* Swarm and Evolutionary Computation, 2022. **72**.

[17] Zou, Y., et al., *An evolutionary algorithm based on dynamic sparse grouping for sparse large scale multiobjective optimization.* Information Sciences, 2023. **631**: p. 449-467.

[18] Chandra, A.a.Y., Xin, *DIVACE: Diverse and accurate ensemble learning algorithm.* Lecture Notes in Computer Science, 2014.

[19] Tian, Y., et al., *PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum].* IEEE Computational Intelligence Magazine, 2017. **12**(4): p. 73-87.

[20] Sun, Y., G.G. Yen, and Z. Yi, *IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems.* IEEE Transactions on Evolutionary Computation, 2019. **23**(2): p. 173-187.

[21] Deb, K. and H. Jain, *An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints.* IEEE Transactions on Evolutionary Computation, 2014. **18**(4): p. 577-601.

[22] Zhang, X., et al., *A competitive mechanism based multi-objective particle swarm optimizer with fast convergence.* Information Sciences, 2018. **427**: p. 63-76.

[23] Derrac, J., et al., *A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms.* Swarm and Evolutionary Computation, 2011. **1**(1): p. 3-18.