Supplemental File of "A Sparsity Learning-based Evolutionary Algorithm for Large-scale Sparse Multiobjective Optimization"

Yongfeng Li[1], Lingjie Li[2,*], *Member, IEEE*, Qiuzhen Lin[1], *Member, IEEE*, Zhong Ming[1,3], and Victor C.M. Leung[1,4,5], *Life Fellow, IEEE*

[1] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2] Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen 518123, China
[3] College of Big Data and Internet, Shenzhen Technology University, Shenzhen 518118, China.
[4] Artificial Intelligence Research Institute, Shenzhen MSU-BIT University, Shenzhen 518172, China
[5] Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver V6T 1Z4, Canada
*: Corresponding Authors

1 Tables

Table 1: Summary of the method descriptions of representative large-scale sparse multiobjective evolutionary algorithms

Algorithms	Year	Method description
SparseEA	2020	Propose the population initialization strategy + binary evolutionary operators
MOEAPSL	2021	Propose the RBF and DAE to reduce the search space of binary decision variables and real variables
PMMOEA	2021	Propose the evolutionary pattern mining approach to reduce the search space of binary decision variables
SECSO	2022	Propose CSO+ Propose convex sparse operator
SNSGAII	2024	Propose sparse uniform initialization strategy +sparse SBX +sparse PM

Table 2: Parameter Settings of Three Application Problems

Problems	D	Dataset	Objectives	M
SR1	1000	Synthetic [[2]]	minimize the reconstruction error and the sparsity of the signal	2
SR2	2000	Synthetic [[2]]		
SR3	4000	Synthetic [[2]]		
KP1	1000	Knapsack [[4]]	minimize the total weight and maximize the total value	
KP2	2000	Knapsack [[4]]		
KP3	4000	Knapsack [[4]]		
NN1	435	$Statlog_{Australian}$ [3]	minimize the error and complexity of the model	
NN2	505	Climate [3]		
NN3	937	$Statlog_{German}$ [3]		

Table 3: Search Strategies And Parameters Settings Of All Compared Algorithms

Algorithms	Search Strategies	Parameter settings
PMMOEA	Proposed pattern mining approach + SBX +PM	$p_c = 1.0$, $p_m = 1/D$, $n_c = 20$, $n_m = 20$
MOEAPSL	Proposed neural network + SBX +PM	$p_c = 1.0$, $p_m = 1/D$, $n_c = 20$, $n_m = 20$
SparseEA	Proposed population initialization strategy + SBX +PM	$p_c = 1.0$, $p_m = 1/D$, $n_c = 20$, $n_m = 20$
SECSO	Proposed CSO+ Proposed convex sparse operator	$\lambda_{initial} = 0.35$, $w = 0.7968$, $c_1 = 1.4962$, $c_2 = 1.4962$
SNSGAII	Proposed population initialization strategy + sparse SBX + sparse PM	$p_c = 1.0$, $p_m = 1/D$, $n_c = 20$, $n_m = 20$
SLEA	Proposed sparsity learning-based mechanism + sparse strategy	$K = 5$

2 Analysis

Here, the analysis of representative LSMOEAs is clarified. Table 1 gives a description on representative large-scale sparse multiobjective evolutionary algorithms. As described in Table 1, representative LSMOEAs handle SMOPs by designing different strategies. However, there still exist some limitations among these LSMOEAs. For example, the designed population initialization strategy in Spars-eEA ensures the sparsity of population, which learns the sparsity of decision variables based on the score of each decision variable before optimization process, while the random competition based on the score for variables may lead to the inaccurate judgment of the sparsity of the variables. Besides, the RBF network model used in MOEAPSL aims to learn the sparse distribution of decision variables from the population during the evolutionary process, however, the population is of poor quality in early evolutionary process, which may result in the limited capacity to learn the sparsity of variables when evaluation cost is low. As for PMMOEA, it uses evolutionary pattern mining approach to mine the sparse distribution of population during optimization process, while the pattern mining method learns the sparse distribution of each decision variable by utilizing evolutionary knowledge of current population essentially. Therefore, if the evaluation resources is limited for the population evolution, the sparsity of decision variables can not be learned sufficiently in PMMOEA. As for the SECSO, it uses convex sparse operator to generate sparse solutions during the evolutionary process, while it needs much evaluation to adjust the sparsity of each decision variable. Consequently, SECSO can not learn the sparsity of variables sufficiently within a small number of evaluations. The proposed varied striped sparse population sampling in SNSGAII ensures the sparsity of population before evolutionary process, which learns the sparsity of decision variables based on the suppose that the sparse variables are distributed in stripes. Therefore, the sparsity of variables may not be judged efficiently if the sparse variables are not distributed in stripes. Based on the above discussions, all LSMOEAs in Table 1 fail to learn the sparsity of decision variables sufficiently, which limits the search speed. Therefore, this motivates us to design a new kind of LSMOEA to enhance the performance.

3 Performance metrics

1) Inverted generational distance (IGD): The smaller IGD value of the algorithm indicates the better performance of algorithm in solving the optimization problem. Specifically, 10000 points evenly sampled from the true ***PF*** of target optimization problem are required for the calculation of IGD. Assume that R is the set of 10000 points uniformly sampled on the true ***PF*** of the given optimization problem and P is a set of the best solutions from corresponding algorithm in solving target optimization problem. Therefore, IGD can be calculated as follows:

$$IGD(P,R) = \frac{\sum_{r \in R} dis(P,r)}{|R|}, \tag{1}$$

where *dis(P, r)* denotes the distance between solutions in P and the reference point r in R, and | | denotes the size of R.

2) Hypervolume (HV): Since true PF in real-world application is unknown, IGD can be not be adopted as the performance indicator. In this paper, HV is adopted as the metric to assess the performance of one algorithm in solving real-world application. The larger HV value of the algorithm indicates the better performance in solving optimization problem. HV is calculated as follows:

$$HV(P) = VoI\left(\bigcup_{x \in P} [f_1(x), z_1] \times ... \times [f_m(x), z_m]\right), \tag{2}$$

where the reference point $z = (z_1, z_2, ..., z_m)$ is set as (1, 1) and P is the final population.

Please Note that all the test problems and real-world problems are run 20 independently times for all algorithms, and the mean and standard deviation of the IGD and HV values are recorded. To ensure a statistically sound conclusion, the Wilcoxon rank sum test with a 0.05 significance level and the Wilcoxon signed-rank test using the platform ***KEEL*** [1] are also used in the experimental analysis. Note that the symbols "+", "-" and "=" in the following tables indicate that compared algorithms are better than, worse than and similar to that of our proposed SLEA, respectively.

4 Experimental Analysis

- Firstly, as for PMMOEA, it performs relatively well in solving SMOP4 and SMOP5 problems with convex shapes since the adopted evolutionary operators ensure the sparsity of population during the evolutionary process, for balancing the exploration and exploitation. However, in most benchmark problems with other shapes, PMMOEA performs worse than our proposed SLEA, as the adopted pattern mining approach can accelerate the convergence speed to some extent, while may result in the local optima.
- Secondly, MOEAPSL performs better than SLEA mainly on SMOP5 problem with unimodality due to the adopted dimension reduction technique in MOEAPSL, which can accelerate the search speed. However, in other problems, MOEAPSL performs worse than SLEA. Moreover, the final population obtained by MOEAPSL has the worse distribution than SLEA. The reason behind this is that the adopted samples training method limits the search in the neighbourhood of current solutions, which may lead to the local optima.
- Thirdly, the performance of SparseEA is better than our proposed SLEA mainly on SMOP5 problem. The reason behind this may be that the designed binary operators improve the search ability when solving unimodal problems. However, SparseEA obtains the poor performance in most problems. The main reason for the poor performance of SparseEA is that the random competition in the importance of variables may incur the slow convergence speed.
- Fourthly, SECSO performs poorly in all problems. The main reason for the poor performance of SECSO can be attributed that tri-particle mechanism based CSO converges slowly within limited evaluation resources, and in this paper, the small amount of evaluation is set for the termination. Consequently, it is hard for SECSO to exhibit the superior performance.
- Fifthly, SNSGAII is outperformed by SLEA mainly on unimodal SMOP5 problem. The reason behind this may be that the proposed evolutionary operators enhance the search efficacy in solving unimodal problems. However, SNSGAII performs worse than SLEA in other problems with multimodality and deception as the striped sparse population initialization strategy may lead to the slow convergence speed.

References

1. Alcalá-FdezJ., SánchezL., GarcíaS., Del, J.J., VenturaS., Garrellj., M., OteroJ., RomeroC., BacarditJ., Rivasv., M.a.: KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing - A Fusion of Foundations, Methodologies and Applications (2008)
2. Liang, J., Gong, M., Li, H., Yue, C., Qu, B.: Problem definitions and evaluation criteria for the cec special session on evolutionary algorithms for sparse optimization. Technical Report, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China, Report# 2018001 (2018)

3. Tian, Y., Zhang, X., Wang, C., Jin, Y.: An evolutionary algorithm for large-scale sparse multiobjective optimization problems. IEEE Transactions on Evolutionary Computation **24**(2), 380–393 (2020). https://doi.org/10.1109/TEVC.2019.2918140
4. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE transactions on Evolutionary Computation **3**(4), 257–271 (1999)