Report of CEC' 2023 Competition on Multiobjective Neural Architecture Search

1. Participant Information

Gui Li
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
Email: liguiatqingdao@gmail.com

Guining Zhan
School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
Email: M202173125@hust.edu.cn

2. Description of the Algorithm

Inspiration: For this neural architecture search benchmark suite, we consider most problems contain fewer objectives, so most of the existing work should be able to maintain a good performance in terms of convergence. Thus, we focus the main work on diversity preservation. Based on our previous work of dynamic learning evolutionary algorithm (DLEA)[1], we add a niche-based diversity maintenance strategy to enhance the diversity. We call this dynamic learning evolutionary algorithm with niche-based diversity maintenance strategy (DLEA-Niche). The following describes the DLEA process.

As shown in Algorithm 1, the population *P* is first initialized (Line 1), and then the iterative update process begins. On the basis of satisfying the iteration condition, tournament selection is performed firstly (Line 3), and the newly generated offspring *Offs* is generated by crossover operator and mutation operator (Line 4). Finally, population *P* and offspring *Offs* are selected through dynamic learning strategy to obtain the new population P_{new} with population size *N*.

Algorithm 1 Framework of DLEA

Input: Population size: N; The number of objective: M; Parameter of p-norm: p
Output: The final population : P

1: $P = Initialization()$
2: **while** $not \quad terminated$ **do**
3: $\quad MatingPool = TournamentSelection()$
4: $\quad Offs = Variation()$
5: $\quad P = EnvironmentalSelection()$
6: **end while**

The environmental selection of DLEA can be found in Algorithm 2. Here, the input parameter P_{2N} refers to the combined population of *P* and *Offs* in Algorithm 1, whose population size is $2\times N$. The purpose is to select *N* best individuals (P_{new}) from the combined population P_{2N}.

As shown in Line 1 of Algorithm 2, the non-dominated sorting of the population P_{2N} is performed to obtain the non-dominated layer *FrontNo* and the maximum layer (the layer requiring the selection individuals) *MaxFNo*. Then, individuals whose number of layers is less than *MaxFNo* are put into P_{new} (Line 2). The convergence-related variables C_n and diversity-related variables D_n are calculated to select individuals in layer *MaxFNo*. The calculation of C_n can be shown in Eq. (1).

$$C_n = \left\lceil nd_i \times \alpha \times (1 - \frac{FEs}{max\,FEs}) \right\rceil \tag{1}$$

where nd_i is the number of individuals that need to be selected in the layer *MaxFNo* at generation *i*. $\alpha \in (0, 1)$ is a convergence factor to control the convergence rate of the algorithm and 0.9 is recommended, the outermost sign is the integer function. *FEs* is the current function evaluations, and *maxFEs* is the preset maximum function evaluations. The calculation of D_n can be shown as follows.

$$D_n = nd_i - C_n \tag{2}$$

The next step calculates the $I_{\varepsilon+}$ [2] of all individuals in population P_{2N} (Line 6), which is the minimum distance required to describe a solution in the objective space in order to dominate another solution, as shown in Eqs. (3).

$$I_{\varepsilon+}(x_1, x_2) = \min_{\varepsilon}(f_i(x_1) - \varepsilon \le f_i(x_2), 1 \le i \le M) \tag{3}$$

In addition, Eq. (4) is used in [2] as the cost required to delete an individual in population, which is equivalent to fitness function.

$$F(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I_{\varepsilon+}(x_2, x_1)/0.05} \tag{4}$$

It is mainly used to measure the convergence of individuals in a population. Here, we use $I_{\varepsilon+}$ as the indicator to select convergence-related individuals in the layer *MaxFNo*. Firstly, we select C_n individuals and put it into P_{new} by this indicator (Lines 9-11). In order to maintain the diversity, the L_p-norm-based diversity maintenance mechanism[2] is used(Line 8). The L_p-norm-based distance has been proved to be superior to Euclidean distance or Manhattan distance as the distance measurement method in diversity maintenance strategy. According to [2], the value p in L_p-norm-based distance is recommended to be set as $1/M$, and M is the objective number. After calculating this distance, we select D_n individuals in the layer *MaxFNo* and put them into P_{new} (Lines 12-14).

Algorithm 2 Environmental Selection

Input: Population: P_{2N}; Population size: N; Current iteration: gen; Maximum iteration: $maxgen$; Parameter of p-norm: p

Output: Next generation population : P_{new}

1: $[FrontNo, MaxFNo] = NDSort()$
2: $P_{new} = P_{2N}(FrontNo < MaxFNo)$
3: Calculate Cn by Eq. (1)
4: Calculate Dn by Eq. (2)
5: //Calculate the $I_{\varepsilon+}$ indicator of individuals
6: $I_{\varepsilon+} = IndicatorI_{\varepsilon+}()$
7: //Calculate the $L_p - norm$ distance between individuals
8: $PNormDis = IndicatorDis()$
9: **for** $i = 1 : Cn$ **do**
10:     Choose the individual by $I_{\varepsilon+}$ and put into P_{new}
11: **end for**
12: **for** $i = 1 : Dn$ **do**
13:     Choose the individual by $PNormDis$ and put into P_{new}
14: **end for**

It is worth noting that the framework of DLEA will continue to select other individuals through dynamic learning strategy after a non-dominated sorting. Therefore, the algorithm can not only adapt to many-objective optimization problems, but also play the important role of non-dominated sorting in multi-objective optimization problems. At the same time, for this competition, we further improved the diversity maintenance ability, which is described in detail as follows:

Inspired by niche-based diversity maintenance strategy in bi-criterion evolution (BCE)[3], we introduce it into DLEA-Niche. The idea is to share resources and measure how crowded an individual is in a population. This strategy is mainly used to truncate the population when there is a large number in the population so that the remaining individuals can maintain a good diversity. The crowding degree of individual p in a population P is defined as follows:

$$D(p) = 1 - \prod_{q \in P, q \neq p} R(p, q) \tag{5}$$

$$R(p, q) = \begin{cases} d(p, q)/r, & \text{if } d(p, q) \leq r \\ 1, & \text{otherwise} \end{cases} \tag{6}$$

where, $d(p, q)$ represents the Euclidean distance between p and q, r is the niche radius and is set as 3 as recommended. Since DLEA reasonably allocates computational (function evaluation) resources according to the iterative situation, the niche technique is very suitable for embedding the diversity preservation part of DLEA to improve diversity.

Different from DLEA, DLEA-Niche uses a niche-based diversity maintenance strategy in environment selection to select diverse individuals (lines12-14, Algorithm 2). It is worth noting that niche techniques are used in BCE to truncate excess non-dominated individuals in PC populations to maintain good diversity. In the dynamic learning strategy, the process is to select the more diverse individuals among the remaining individuals after the non-dominated sorting and the convergent individual selection. Although niche techniques may not play much of a role in the early stage of the algorithm, this is due to the early stage of dynamic learning strategies leaving computational (functional evaluation) resources to convergent individuals. As the algorithm iterates, this center of gravity will shift. More computing resources will be devoted to maintaining diversity, allowing niche-based diversity maintenance strategy to make a huge difference. This process is also an advantage of dynamic learning strategy. By this method, the diversity of the algorithm is effectively guaranteed. The experiment in the third part effectively verifies the performance improvement.

3. Experiments on NAS Benchmark Suites

We did 31 independent runs of each benchmark problem according to the parameters set by the competition. The parameter settings are shown in Table 1.

Table 1 Parameter Setting

Problem	N	M	D	FE	runs
C10MOP1	100	2	26	10000	31
C10MOP2	105	3	26		
C10MOP3	105	3	5		
C10MOP4	120	4	5		
C10MOP5	126	5	6		
C10MOP6	132	6	6		
C10MOP7	156	8	6		
C10MOP8	100	2	32		
C10MOP9	105	3	32		
IN1KMOP1	100	2	25		
IN1KMOP2	100	2	25		
IN1KMOP3	105	3	25		
IN1KMOP4	100	2	34		
IN1KMOP5	100	2	34		
IN1KMOP6	105	3	34		
IN1KMOP7	100	2	21		
IN1KMOP8	100	3	21		
IN1KMOP9	120	4	21		

The experimental results are shown in Table 2-4. Table 2 and 3 show the HV results of 31 runs of DLEA and DLEA-Niche on the C10MOP and IN1KMOP benchmarks respectively. Table 4 shows the IGD results of 31 runs of the two algorithms on C10MOP1-7. The two values in the table represent the mean and standard deviation respectively (in parentheses). The symbols +, -, and = stood for that the indicator values of the DLEA significantly better than, worse than, and similar to that of DLEA-Niche (under rank-sum test).

Table 2 HV result of DLEA and DLEA-Niche runs on C10MOP benchmark suite

Problem	DLEA	DLEA-Niche
C10MOP1	**9.3894e-1 (6.28e-3) =**	9.3766e-1 (6.18e-3)
C10MOP2	9.1637e-1 (2.58e-3) =	**9.1647e-1 (4.71e-3)**
C10MOP3	8.2630e-1 (1.25e-3) -	**8.2884e-1 (4.07e-3)**
C10MOP4	7.6495e-1 (1.16e-2) -	**7.8114e-1 (5.88e-3)**
C10MOP5	7.1239e-1 (2.44e-4) =	**7.1244e-1 (3.29e-4)**
C10MOP6	**7.4027e-1 (1.12e-4) +**	7.4025e-1 (4.87e-4)
C10MOP7	**5.9029e-1 (1.01e-2) =**	5.8386e-1 (4.22e-3)
C10MOP8	9.4548e-1 (7.16e-3) -	**9.7086e-1 (5.00e-3)**
C10MOP9	9.2661e-1 (1.12e-2) -	**9.5231e-1 (1.39e-2)**
+/-/=	1/4/4	

Table 3 HV result of DLEA and DLEA-Niche runs on IN1KMOP benchmark suite

Problem	DLEA	DLEA-Niche
IN1KMOP1	**9.1253e-1 (6.41e-3) =**	9.0923e-1 (7.55e-3)
IN1KMOP2	8.7629e-1 (3.60e-3) -	**8.7898e-1 (1.90e-3)**
IN1KMOP3	8.0311e-1 (4.82e-3) -	**8.0946e-1 (5.78e-3)**
IN1KMOP4	9.7561e-1 (2.10e-2) =	**9.8020e-1 (6.13e-3)**
IN1KMOP5	9.8464e-1 (8.95e-3) =	**9.8472e-1 (6.80e-3)**
IN1KMOP6	**9.6641e-1 (9.65e-3) =**	9.6539e-1 (8.94e-3)
IN1KMOP7	**8.4959e-1 (2.21e-2) =**	8.4429e-1 (2.78e-2)
IN1KMOP8	6.9662e-1 (1.33e-2) -	**7.1180e-1 (8.26e-3)**
IN1KMOP9	6.1262e-1 (1.23e-2) -	**6.3196e-1 (5.48e-3)**
+/-/=	0/4/5	

Table 4 IGD result of DLEA and DLEA-Niche runs on C10MOP1-7

Problem	DLEA	DLEA-Niche
C10MOP1	3.2522e-2 (1.03e-2) =	**3.2096e-2 (9.64e-3)**
C10MOP2	3.2472e-2 (8.35e-3) =	**2.9704e-2 (9.04e-3)**
C10MOP3	2.6731e-2 (1.84e-3) -	**2.1267e-2 (2.71e-3)**
C10MOP4	5.9207e-2 (2.49e-3) -	**5.3476e-2 (1.87e-3)**
C10MOP5	2.1301e-2 (1.01e-2) =	**2.0051e-2 (1.12e-2)**
C10MOP6	1.3691e-2 (5.40e-3) -	**1.0673e-2 (7.34e-3)**
C10MOP7	7.2196e-2 (7.30e-3) -	**3.3219e-2 (5.45e-3)**
+/-/=	0/4/3	

According to the HV result of C10MOP benchmark problems, the result of DLEA-Niche is significantly improved compared with DLEA, except for C10MOP9. The improvement in other problems is obvious. Six optimal results were maintained in nine test cases. Similar to the results from C10MOP, the performance of the two algorithms on IN1KMOP was consistent. Both sets of results demonstrated the improved performance of DLEA-Niche. Finally, the IGD results on 7 problems with known Pareto front calculated by DLEA-Niche performed significantly better than DLEA.

Please note that the versions used in this experiment are **EvoXBench1.0.3**, **pymoo0.6.0.1** and **PlatEMO4.1**. The hardware environment of the experiment is **i7-12700** and the system version is **Windows 11**. Other experimental data support files (JSON) and algorithm source code are placed in the "**.zip**" file, please check.

References

[1] G. Li, G.-G. Wang, J. Dong, W.-C. Yeh, and K. Li, "DLEA: A dynamic learning evolution algorithm for many-objective optimization," *Information Sciences,* vol. 574, pp. 567-589, 2021.

[2] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation,* vol. 19, no. 4, pp. 524-541, Aug, 2015.

[3] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation,* vol. 20, no. 5, pp. 645-665, 2015.