Report of CEC' 2024 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 improved the dynamic learning evolutionary algorithm (DLEA) [1] in last year's CEC2023 competition, called DLEA-Niche. In the DLEA-Niche. We added a niche-based diversity maintenance strategy to enhance diversity. And in CEC2024 competition, we continue to make improvements based on the flaws of DLEA-Niche and characteristics of problems. A robust optimization strategy and extreme points retention strategy are proposed. The following describes the DLEA process and Improvement strategy.

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 environmental selection to obtain the new population P_{new} with population size N. Among them, dynamic learning strategy is used in environment selection, which is also the key point.

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 terminated do
3:     MatingPool = TournamentSelection()
4:     Offs = Variation()
5:     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 of environmental selection is to select N best individuals from the combined population P_{2N} as the parent population (P_{new}) of the next iteration.

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 number of 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. (2).

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

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. And 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 the use of 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 them 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:

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

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

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 is an algorithm that 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 big difference. This process is also an advantage of dynamic learning strategy. By this method, the diversity of the algorithm is effectively guaranteed.

Considering that the dynamic objective value of NAS problem, we also propose a robust optimization strategy. The goal is to achieve robustness gains by sacrificing a small number of function evaluations. First, we sample the maximum number of iterations t times. The interval of t sampling is different, and this effect is achieved by an exponential function. When the number of iterations is satisfied, we evaluate the objective value twice for the population, and finally use the minimum value of the two objectives for environment selection. This method can achieve more robust results by sacrificing a small number of function evaluations. And the improvements in execution robustness are more frequent later in the iteration. To ensure that only a small amount of function evaluation is sacrificed, t is set to 5 in our algorithm.

In addition, we found that although niche-based diversity maintenance strategy can ensure better diversity, they cannot guarantee the preservation of extreme points. The extreme points are crucial to ensure the diversity. Therefore, we consider adding an extreme points retention strategy at the end of the environment selection. After the previous environment selection is completed, we will identify the M extreme points. If any extreme point is not preserved, then we add the extreme point to the next generation. At the same time, the strategy will also delete the selected solution with the minimum distance from the newly added extreme point.

We apply robust optimization strategy and extreme points retention strategy to improve DLEA-Niche (DLEA-NRE). The experimental results in the third part verify the superiority of DLEA-NRE.

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
CitySegMOP1	100	2	24		
CitySegMOP2	105	3	24		
CitySegMOP3	105	3	24		
CitySegMOP4	120	4	24		
CitySegMOP5	126	5	24		
CitySegMOP6	100	2	24		
CitySegMOP7	105	3	24		
CitySegMOP8	105	3	24	10000	31
CitySegMOP9	120	4	24		
CitySegMOP10	126	5	24		
CitySegMOP11	105	3	24		
CitySegMOP12	126	5	24		
CitySegMOP13	132	6	24		
CitySegMOP14	132	6	24		
CitySegMOP15	217	7	24		

Table 2 shows the HV results of 31 runs of DLEA-Niche and DLEA-NRE on the CitySegMOP benchmarks respectively. 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-Niche significantly better than, worse than, and similar to that of DLEA-NRE.

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

Problem	N	M	D	FE	DLEA-Niche	DLEA-NRE
CitySegMOP1	100	2	24	10000	8.9900e-1 (1.35e-3) =	**9.0008e-1 (1.65e-3)**
CitySegMOP2	105	3	24	10000	**7.9890e-1 (1.59e-3) =**	7.9870e-1 (1.20e-3)
CitySegMOP3	105	3	24	10000	8.2144e-1 (4.43e-3) =	**8.2237e-1 (2.81e-3)**
CitySegMOP4	120	4	24	10000	6.9532e-1 (5.85e-3) =	**6.9735e-1 (1.66e-3)**
CitySegMOP5	126	5	24	10000	6.5585e-1 (8.87e-4) =	**6.5601e-1 (8.90e-4)**
CitySegMOP6	100	2	24	10000	7.7122e-1 (3.85e-3) =	**7.7190e-1 (2.84e-4)**
CitySegMOP7	105	3	24	10000	**7.3125e-1 (1.22e-4) =**	7.3120e-1 (1.43e-4)
CitySegMOP8	105	3	24	10000	7.2927e-1 (1.73e-2) =	**7.3240e-1 (1.90e-4)**
CitySegMOP9	120	4	24	10000	5.7635e-1 (1.69e-3) =	**5.7666e-1 (4.16e-4)**
CitySegMOP10	126	5	24	10000	5.4714e-1 (3.97e-4) =	**5.4716e-1 (3.38e-4)**
CitySegMOP11	105	3	24	10000	6.8581e-1 (5.78e-3) =	**6.8598e-1 (5.14e-3)**
CitySegMOP12	126	5	24	10000	4.7056e-1 (2.11e-3) -	**4.7182e-1 (1.15e-3)**
CitySegMOP13	132	6	24	10000	4.2109e-1 (1.71e-3) -	**4.2193e-1 (1.19e-3)**
CitySegMOP14	132	6	24	10000	4.3276e-1 (1.85e-3) -	**4.3372e-1 (8.79e-4)**
CitySegMOP15	217	7	24	10000	**3.9821e-1 (7.87e-4) =**	3.9810e-1 (6.74e-4)

According to the HV result of CitySegMOP benchmark problems, the result of DLEA-Niche is improved compared with DLEA, except for CitySegMOP2, CitySegMOP7, and CitySegMOP15. Twelve optimal results were maintained in Fifteen test cases.

Please note that the versions used in this experiment are **EvoXBench1.0.5** and **PlatEMO4.7**. 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.