Improved Evolutionary Operators Based on Different Encoding Regions

Pengcheng Jiang[1], Chenchen Zhu[1] and Yu Xue[1]

(1. School of Software, Nanjing University of Information Science and Technology, Jiangsu 210044, China. Email: pcjiang@nuist.edu.cn; zhuchenchen0411@163.com; xueyu@nuist.edu.cn)

1、Introduction

Architectures from different search spaces are often encoded in different ways, and there may be different encoding regions for individuals within a search space. Different encoding methods will affect the subsequent search process and the search results. In the original paper, the same evolutionary operators were used for the different encoding regions. But these encoding regions have different characteristics, and the evolutionary algorithm best suited to each encoding region should also be different. So, we improve the evolutionary algorithm in the neural architecture search process. According to the encoding properties of each search space, corresponding sampling methods, crossover and variation operators are designed for different encoding regions.

2、Proposed Method

2.1 Design of Evolutionary Operators for Different Encoding Regions

2.1.1 Binary-based Encoding Region

Individuals based on binary encoding are represented by 0 and 1, and the corresponding evolutionary operators are designed. First, when sampling individuals, binary sampling method is used. Individuals are selected according to the distribution probability of 0-1 in each individual. For example, the probability is set to 0.5. When sampling every bit of an individual, if the random value is greater than 0.5, the bit is 1; if the random value is less than 0.5, this bit is 0. Then, mutation operator is the bit flipping mutation operator, and each bit of individual code is mutated to 0 or 1 according to the mutation probability. If the random value is greater than the mutation probability, it will flip from 0 to 1 or 1 to 0. The crossover operator uses uniform crossover. Each bit of the two parent individuals is exchanged according to the crossover probability, thus forming a new individual.

2.1.2 Real Number-based Encoding Region

For real number-based encoding region, each bit of encoding has a certain candidate range. For the encoding area based on real numbers, individuals are sampled using choice sampling. Each bit of an individual randomly selects a value according to the candidate range, and then generates an individual. Then mutation operator is the choice mutation, each bit of the individual code is mutated within the candidate range according to the mutation probability. Finally, two-point crossover is chosen to used, two crossover points are randomly set in two randomly selected parent individuals, and then the codes between the two crossover points are exchanged.

2.1.3 Numerical Size-based Encoding Region

The encoding area is encoded based on the size of the values, with the minimum and maximum values set as the upper and lower bounds, and each value corresponding to a sequence number. For sampling individuals in this coding area, Latin hypercube sampling is used. The sampling range is divided into layers, and a random value within a candidate range is selected for each layer to generate a sampled individual. The polynomial mutation operator is selected for mutation. Individuals undergo polynomial mutation according to Eq1.

$$\begin{cases} C = [2 \times \mu + (1 - 2 \times \mu)(1 - \delta_1)^{n+1}]^{\frac{1}{n+1}} - 1 & \mu \leq 0.5 \\ C = 1 - [2 \times (1 - \mu) + 2 \times (\mu - 0.5)(1 - \delta_2)^{n+1}]^{\frac{1}{n+1}} & \mu > 0.5 \end{cases} \tag{1}$$

Where C is the individual obtained after mutation, μ is a random number between 0 and 1, n is the distribution factor. The simulated binary crossover operator is chosen for crossover. Simulated binary crossover is performed according to Eq2.

$$\begin{cases} C_1 = \frac{1}{2}[(1 + \beta) \times P_1 + (1 - \beta) \times P_2] \\ C_2 = \frac{1}{2}[(1 - \beta) \times P_1 + (1 + \beta) \times P_2] \end{cases} \tag{2}$$

Where P_1、P_2 are two parent individuals, C_1、C_2 are two offspring individuals generated after crossover, β is a random number for the expansion factor.

2.2 Detailed Design of Evolutionary Operators for Various Search Spaces

2.2.1 NASBench-101

The individual is composed of a directed acyclic graph with up to 7 nodes representing operations besides input and output nodes. The encoding area is divided into two parts, the first 21 bits represent binary coding indicating connectivity between nodes, and the last 5 bits represent candidate operations coded in real numbers ranging from [0,2]. Therefore, for sampling in this search space, binary sampling is used for the first 21 bits, while choice sampling is used for the remaining 5 bits. The mutation operator adopts bit-flip mutation, where each bit in the individual's encoding mutates into 0 or 1 according to the mutation probability. The crossover operator uses uniform crossover, each bit of two parent individuals is exchanged according to the crossover probability, thus forming a new individual.

2.2.2 NASBench-201

The individuals in this search space are composed of four nodes, connected by six edges representing operations. The candidate range for each edge is [0,4]. Individuals in this space are encoded by the numbers of the candidate operations, with a total of six bits. Therefore, individuals in this space are sampled using choice sampling. Six candidate operation are randomly chosen to form an individual. The mutation operator is choice mutation. Each bit of an individual is replaced with another candidate operation randomly based on the mutation probability. The crossover operator is two-point crossover. Two crossover points are randomly set in two randomly selected parent individuals, and then the codes between the two crossover points are exchanged.

2.2.3 NAST-Bench

The individuals in this search space are stacked by five cells or blocks and encoded by the channel number of each module, a total of 5 bits. The candidate range for the channel number is [8,16,24,32,40,48,56,64]. The channel number is numbered sequentially from [0,7] for convenience. Based on the encoding characteristic of numerical values, Latin hypercube sampling is used to sample individuals in this search space. And, mutation operator is polynomial mutation. The crossover operator adopts simulated binary crossover.

2.2.4 DARTS

The individuals in this search space are stacked by Normal Cells and Reduction Cells. The individual encoding consists of two parts, 16 bits for each Normal Cell and Reduction Cell, and 32 bits in total. Normal Cells and Reduction Cells have the same structure which consist of one input node, one output node, and four intermediate nodes. Nodes are directly connected pairwise, with a total of 8 edges, each edge having two candidate operations. Individuals are encoded based on the candidate operation number, within the range of [0,6]. The individuals are sampled by choice sampling to randomly select 32 digits within the candidate value range. The mutation operator is choice mutation. Each bit of the individuals is randomly replaced by another candidate operation based on the mutation probability. The crossover operator is two-point crossover. Two crossover points are randomly set in two randomly selected parent individuals, and then the codes between the two crossover points are exchanged.

2.2.5 ResNet50D

The individuals in this search space are mainly stacked by two basic blocks, Identity Block and Conv Block. Each individual consists of 5 stages, the structure of Stage 0 is simple and can be viewed as the preprocessing of the input. The other four stages are all composed of Bottlenecks and have similar structures. The individuals in this search space are encoded by real-number, which is mainly divided into three parts with a total of 25 bits. The first part is the 0th bit, which represents the image size information encoded by the index of each size in the image size list. The second part is the 1st to 6th bits, representing the channel multiplier information. It is encoded by the value of "w+1". When the depth d[0] is equal to 0, the first two bits are random. The third part represents the information of the 4 stages, which are represented by 4, 4, 6, and 4 bits respectively. The index of the expansion rate in the expansion rate list is used for encoding. The depth list d[1-4] indicates that 2-d[x] bits will miss in each stage. When there is a missing bit, the preceding expansion rate is represented by 0. For this search space is encoded by real-number, choice sampling is used for sampling to randomly select values from the candidate range to generate individuals. The mutation operator uses choice mutation. Each bit of the individual is randomly replaced with another candidate operation based on the mutation probability. The crossover operator is two-point crossover, two crossover points are randomly set in two randomly selected parent individuals, and then the codes between the two crossover points are exchanged.

2.2.6 Transformer

The individuals in this search space are encoded by real numbers and divided into four parts. The first part is the 0th bit, representing the depth. The second part is the 1st bit, representing the hidden layer dimension. The third part's length is influenced by the depth and consists of bits 2 to 2 + the depth value, representing the expansion rate of the hidden layers. The expansion rate index is used for encoding. The fourth part's length is determined by the depth and is from 2 + the maximum value of the depth list to 2 + the maximum value of the depth list + the depth value bits. The attention head index is used for encoding. In this search space, individuals are sampled by choice sampling, randomly selecting values from the candidate range to generate individuals. The mutation operator is choice mutation, and each bit of the individual is replaced with other candidate operations randomly based on the mutation probability. The crossover operator is two-point crossover, two crossover points are randomly set in two randomly selected parent individuals, and then the part between the two crossover points are exchanged.

2.2.7 MobileNetV3

Individuals are encoded by real number with a total of 21 bits in this search space. The first bit represents the information of the image size, which is encoded by the index of the candidate image size in the list of candidate image sizes. The following 20 bits represent information about the five stages. Each stage has a pair of kernel size and expansion rate, which are represented as a single value. There are three candidate kernel sizes and three candidate expansion rates, resulting in nine pairs. In addition, skip connections are included, resulting in ten groups that are represented by [0,9]. for real number encoding, we use choice sampling to randomly select values from the candidate range to generate an individual. The mutation operator is choice mutation, and each bit of the individual is replaced with another candidate operation based on the mutation probability. The crossover operator is two-point crossover. Two crossover points are randomly set in two randomly selected parent individuals, and then the part between the two crossover points are exchanged.

3、Results

In the following, we present results achieved by our method on the two test suites.

1) Results on C-10/MOP: Table I shows the statistical values of the HV metric achieved.

2) Results on IN-1K/MOP: The statistical values of the IGD metric achieved by our method are showed in Table II.

TABLE I: Statistical results (median and standard deviation) of the HV and IGD values on C-10/MOP test suite.

PROBLEM	IGD	HV
C-10/MOP1	0.0657±0.0101	0.8984±0.0134
C-10/MOP2	0.0620±0.0119	0.8756±0.0127
C-10/MOP3	0.0391±0.0020	0.8102±0.0023
C-10/MOP4	0.0686±0.0024	0.7592±0.0066
C-10/MOP5	0.0693±0.0140	0.7081±0.0011
C-10/MOP6	0.0428±0.0074	0.7319±0.0045
C-10/MOP7	0.0791±0.0103	0.5746±0.0108
C-10/MOP8	/	0.9333±0.0105
C-10/MOP9	/	0.9025±0.0182

TABLE II: Statistical results (median and standard deviation) of the HV values on IN-1K/MOP test suite.

PROBLEM	HV
IN-1K/MOP1	0.8866±0.0051
IN-1K /MOP2	0.8393±0.0067
IN-1K /MOP3	0.7901±0.0057
IN-1K /MOP4	0.9108±0.0094
IN-1K /MOP5	0.9071±0.0124
IN-1K /MOP6	0.8974±0.0094
IN-1K /MOP7	0.8375±0.0072
IN-1K /MOP8	0.6547±0.0092
IN-1K /MOP9	0.5748±0.0074