Multi-objective Differential Evolution Algorithm Based on Affinity Propagation Clustering

Dan Qu, Hongyi Li and Huafei Chen

Abstract—**Multi-objective problems have gained much attention during the last decade. To balance the diversity and the convergence of the multi-objective differential evolution algorithm (MODE), an improved MODE is proposed based on the affinity propagation clustering (APC) and the non-dominated count approach in this paper. The proposed algorithm is referred to as AP-MODE, which improves the search efficiency by utilizing the affinity propagation approach to find out the population distribution structure for guiding search. In addition, mating restriction probability is used to select parent individuals for recombination from the neighborhoods or the whole population. Meanwhile, the mating restriction probability is updated according to the non-dominated count approach at each generation. This proposed algorithm is verified by comparing it with some state-of-the-art multi-objective evolutionary algorithms, and the simulation results on DTLZ test problems indicate that AP-MODE can efficiently achieve two goals of multi-objective optimization, i.e., the convergence to actual Pareto front and uniform spread of individuals along Pareto front.**

Index Terms—**Differential Evolution Algorithm, Affinity Propagation, Clustering, Multi-objective Optimization.**

I. Introduction

MULTI-OBJECTIVE optimization problems [1]-[7] are the main topics of optimal decision issues in research and applications. These multiple objectives are often interrelated and mutually conflicted, meaning that any improvement in one objective may lead to the degradation of at least one objective. With the emergence of numerous multi-objective problems, the corresponding optimization algorithms have been continuously improving. Such as non-dominated sorting based genetic algorithm (NSGA-II)[7], the evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach (NSGA-III) [9], multi-objective evolutionary algorithm based on decomposition (MOEA/D) [10], ε-dominance based multi-objective evolutionary algorithm ($\varepsilon - MOEA$)[11], improving the Strength Pareto Evolutionary Algorithm for Multi-objective Optimization (SPEA-2) [12].

Differential evolution (DE) is an efficient evolutionary algorithm introduced by Price and Storn[13]. In contrast to traditional EA, DE employs a series of operations to produce offspring, including variation and crossover of individuals in the current population. DE algorithm also has a low space complexity. Due to its excellent properties, DE algorithms are more conducive to handling large-scale, high-complexity optimization problems and have been successfully applied to solve optimization problems in many scientific and engineering fields [14]-[17]. Meanwhile, many researchers intend to extend it to solve multi-objective optimization problems (MOPs). In 1999, Chang[18] first introduced the dominance relationship among individuals into the DE algorithm and acted as a pioneer in solving MOPs using the DE algorithm. Recently, multi-objective differential evolution algorithms (MODE) are being developed, and many algorithms have been proposed. Abbass et al. [19] first proposed a Pareto Differential Evolution (PDE) algorithm for solving MOPs by incorporating Pareto dominance. PDE was compared with SPEA on two test problems and outperformed it. PDE showed the great potential of DE algorithms in solving MOPs and greatly contributed to the advancement of the field. To improve the convergence of the solution set and the uniformity of the solution distribution, Xue *et al.* [20] proposed MODE, which employs a new Pareto-based ranking assignment and crowding distance metric. MODE was tested on five benchmark problems, and it achieved more favorable outcomes than SPEA. Ali *et al.* [21] introduced a multi-objective differential evolution algorithm that utilizes the advantage of dyadic-based learning to generate an initial population of potential candidates and uses the concept of random localization in the variation step. Li *et al.* [22] proposed a decomposition-based MODE algorithm that uses DE instead of a genetic algorithm as a search engine to improve the diversity and convergence of the solution set. However, the mutation pattern of the DE algorithm has not been further improved. In 2015, Zheng *et al.* [23] designed a multi-objective differential selection strategy based on two populations to ensure the quality of the generated solutions. In order to prevent existing MODEs from falling into the local optimum, jumping gene operation was introduced, and a novel MODE based on jumping genes was presented by Liu[24]. The phenomenon of jumping

Manuscript received March 25, 2023; revised October 5, 2023.

This work was supported by the Opening Project of Sichuan Province University Key Laboratory of Bridge Non-destruction Detecting and Engineering Computing. (2021QYY03, 2021QYY02, 2020QYY02).

Dan Qu is a teaching assistant at the College of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong Sichuan 643002, P. R. China (corresponding author, phone: 13980748735; fax: 13980748735; e-mail: 13980748735@163.com).

Hongyi Li is a teaching assistant at the College of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong Sichuan 643002, China(e-mail: 1261234217 @qq.com).

Huafei Chen is a teaching assistant at the College of Mathematics and Statistics, Sichuan University of Science & Engineering, Zigong Sichuan 643002, China(e-mail: huafeichen0502@163.com).

genes reveals that jumping genes on chromosomes can jump from one position to another. The algorithm has been proven to be highly effective and exhibits excellent convergence performance. Li *et al.* [25] proposed a modified differential evolution method with a self-adaptive parameters method, and the proposed algorithm adopted two variation rules based on the rand and the best individual in the whole population. Ai et al. [26] proposed an improved multi-objective differential evolutionary algorithm with archive and spherical pruning based on the MODE-MSRM algorithm, which improves the convergence and robustness of the algorithm. Li *et al.* [27] proposed a MODE algorithm, which was formed by incorporating the memory mechanism of PSO. The personal best concept is utilized in MODE to memorize the set of non-dominated solutions found by each solution. In 2022, Yu et al. [28] developed a constrained multi-objective differential evolution algorithm with a ranking mutation operator, which can find a well-distributed Pareto front. Tian Ye[29] team designed a hybrid algorithm based on differential evolution and a conjugate gradient method tailored for large-scale multi-objective optimization problems (LSMOPs). Compared to state-of-the-art MOEAs and classic algorithms, the proposed algorithm performs better for solving LSMOPs.

In summary, MODE is proven to be a simple and effective algorithm for solving MOPs. Although many methods for improving MODE are proposed constantly, MODE is still facing challenges. In order to improve its performance further, this paper proposed a new approach to extend DE to make it suitable for solving MOPs, which is called AP-MODE (Multi-Objective Differential Evolutionary Algorithm based on Affinity Propagation). Its details are introduced in the following section. The main contributions of this work can be enumerated as follows:

1) Aiming at guiding the search process and accelerating convergence, and affinity propagation clustering is utilized to discover the structure of the Pareto solution set.

2) For population crossover mutations, an adaptive method of recombination and mating restriction probability is proposed to make a population more universal and increase the diversity of the population.

3) The proposed algorithm is verified through a series of comprehensive experiments on benchmark instances, and The results show that the proposed algorithm has better convergence and diversity than the competitor algorithms.

This paper is organized as follows: Section II describes the AP approach and the new DE strategy. Section III describes the framework of the proposed algorithm and its detailed components. Section IV shows the performance comparison. Finally, some conclusions are displayed in Section V.

II. BASIC IDEAS AND CONCEPTS

The MOP[30] considered in this paper is formulated as follows:

$$\min F(x) = (f_1(x), f_2(x), ..., f_m(x))^{\mathrm{T}} \qquad (1)$$

where $x = (x_1, ..., x_n)^{\mathrm{T}} \in \Omega$ is the *decision vector* of dimension n and $\Omega \in R^n$ is *feasible space*. The image set, $S = \{F(x) \mid x \in \Omega\}$, is called *objective space*.

A. DE

DE was introduced by Storn in 1995. The classical DE algorithm contains four main steps during the optimization process: initialization, mutation, crossover, and selection. The algorithmic details of the four steps are outlined in Algorithm 1. There are three main control parameters that significantly affect the performance of DE, which contain population size (NP), crossover rate (C_r), and mutation scale factor (F_{DE}). It is worth noting that the differential mutation operators may have different search behaviors. It is worth noting that the differential mutation operators may have different search behaviors. Four basic most commonly used mutation operators that are listed as follows:

DE/rand/1: $V_i = X_{r1} + F_{DE}(X_{r2} - X_{r3})$;

DE/best/1: $V_i = X_{best} + F_{DE}(X_{r2} - X_{r3})$;

DE/rand/2: $V_i = X_{r1} + F_{DE}(X_{r2} - X_{r3}) + F_{DE}(X_{r4} - X_{r5})$;

DE/current-to-best/1:

$V_i = X_i + F_{DE}(X_{best} - X_i) + F_{DE}(X_{r1} - X_{r2})$

Where V_i denotes the generated perturbed vector, $F_{DE} \in (0,1]$ is the control parameter, which mainly affects the global optimization ability of the algorithm. The smaller the value F_{DE} is, the better the local search ability of the algorithm will be. The algorithm has a greater probability of escaping the local minimum with a larger F_{DE} value. However the convergence speed will be slower. In addition, F_{DE} also affects population diversity. X_{best} is the random non-dominated solution from the global external archive. $X_{r1}, X_{r2}, X_{r3}, X_{r4}, X_{r5}$ are different solutions randomly selected from the current population.

Algorithm 1 Outline of DE's main procedure

Step 1. Initialize population P randomly and evaluate the individuals of P .
Step 2. While the stopping criterion is not satisfied, do:
2.1 for P_i ($i = 1, ..., NP$) in P repeat:
(2.1.1) Create a candidate from the parent P_i
(2.1.2) Evaluate the candidate
(2.1.3) If the candidate individual is better than the parent, the candidate replaces the parent. Otherwise, discard the candidate.
2.2 Randomly enumerate the individuals in P

B. MODE

Being a powerful heuristic for numerical optimization, DE has been extended into MOPs by many researchers. However, there exist many difficulties. On the one hand, pursuing a diverse Pareto front is a challenging issue for any MOEA. On the other hand, deciding how to deal with parent and candidate solutions is difficult. For these considerations, MODE adopts the following principles:

1) If the candidate solution outperforms ($\prec$) the parent solution, the candidate solution replaces it.

2) If the parent solution outperforms ($\prec$) the candidate solution, the candidate solution is rejected.

3) Otherwise (when there is no dominance relationship between them), the candidate solution is added to the current population.

Now, the population size is between NP and $2 \times NP$. Then, the current population is truncated to prepare for the next step of evolution. The so-called truncation comes from NSGA-II, which selects the best NP individuals to form a new population based on ranking and crowding distance. The procedure of MODE is described in Algorithm 2.

Algorithm 2 Outline of MODE

Input: max generation $G_{\max}$; population size NP
Output: Pareto optimal solutions
Step 1. Initialize population $P=\{x_1,x_2,...,x_{NP}\}$
Step 2. While $t<G_{\max}$ (t is current generation), do:
2.1 For each individual of P repeat:
(a) Create candidate solution from parent P (i.e. Solution generation)
(b) Evaluate candidate solution
(c) If the candidate solution dominates the parent, replace the parent with the candidate solution. If the parent dominates the candidate solution, discard the candidate solution. Otherwise, add a candidate solution to the population.
2.2 If the population size exceeds NP , truncate it using the environmental selection approach.
2.3 Randomly enumerate solutions in P .
2.4 Let algebraic counter $t=t+1$.

C. Affinity propagation

Clustering methods[31]-[33] divide objects into groups or clusters based on their similarity in some attributes. The clustering approach aims to make objects within a class as similar as possible and objects between classes as different as possible. The specific steps of the AP[34]-[35] algorithm is depicted in the following algorithm 3:

Algorithm 3 Outline of AP's main procedure

Input: Data: points to be clustered, $G_{\max}$: Maximum iterative generations
Output: the cluster to which each data point belongs
Step 1. Initialization: generation counter $t=0$; reference degree $p(k)$; similarity matrix S ;
Step 2. main loop:
2.1 Calculate matrix $A=[a(i,k)]_{N\times N}$ and $R=[r(i,k)]_{N\times N}$
2.2 Update the matrix A and R according to following formulas:

$$r^t(i,k)=(1-\lambda)*r^t(i,k)+\lambda*r^{t-1}(i,k)$$

$$a^t(i,k)=(1-\lambda)*a^t(i,k)+\lambda*a^{t-1}(i,k)$$

2.3 Determine the center of the points
2.4 Distribution of each point
2.5 $t=t+1$
Step 3. Stop iteration: if $t>G_{\max}$, print out result, otherwise, turn to Step 2.

In AP-MODE, for each solution $x_i\in P_t$, its parent can originate in two sources: the first one is the solution in a cluster containing x_i , which is good at generating high-quality solutions to promote exploitation, and the second one is the whole population P_t , which can enhance exploration. One of the most important aspects to be discussed is how parental individuals are selected from two sources to achieve a balance between exploration and exploitation. At the early evolution stage, the first individual can quickly lose diversity. In the later generation, the other source performs poorly in generating better solutions. Hence, an adaptive parameter β is employed to balance the exploitation and exploration. This parameter β determines which parents are selected from which source. The parents originate from the neighborhoods with probability β ; the parents are selected from the entire population. (i.e., mating restriction) with probability $1-\beta$. In practice, many high-quality solutions are desirable at the early evolutionary stage, while the latter evolutionary stage should focus on stimulating the diverse spread of solutions. Therefore, the mating restriction probability β may also require different values at different evolutionary stages. In order to achieve the above goals, the non-dominated count-based approach is proposed to adjust β adaptively at each generation, which is shown in (2):

$$\beta_{t+1}=\frac{|F_t|+\varepsilon}{|F_t|+NP+\varepsilon} \qquad (2)$$

In this formula, NP is the population size, $|F_t|$ refers to the number of non-dominated solutions that rank in the first front, and $\varepsilon=10^{-10}$ is set to ensure the rationalization of this design.

D. Part and Select Algorithm (PSA)

The Part and Select Algorithm (PSA), developed by Salomon et al. [36], is capable of selecting well-spread points from a given set of points in the objective space that contains potential solutions. The minimal requirements of PSA make it suitable for use as a selection mechanism and/or as a crowding assignment mechanism, which can improve the diversity of the Pareto front. It is recommended to refer to [37]-[38] for a clear introduction to the PSA.

III. IMPROVED MODE

MODE is widely used to solve multi-objective problems (MOPs). There are two essential parts in the multi-objective evolutionary algorithm: variation and environmental selection. These two parts contribute to MODE ' s performance. However, most existing research prioritizes the selection operator over the reorganization operator[39]. Thus, many multi-objective evolutionary algorithms (MOEA) directly use recombination operators designed for single-objective evolutionary algorithms. However, there are essential differences between single-objective evolutionary algorithm and multi-objective evolutionary algorithm in terms of the topological structure of the solution, i.e., the former is one or several independent points, while the latter is a Pareto solution set with regular flow pattern structure. Therefore, it is necessary to redesign the specific recombination operator considering the practical characteristics of MOPs. For these considerations, this improved algorithm denoted by AP-MODE is proposed, which takes MODE as the framework, introducing the affinity propagation clustering and non-dominated counter adaptive mating restriction approach. After the initialization process, the population is clustered by the affinity propagation method, and a new solution by Solution Generation is generated, thus accelerating the exploration and exploitation of the whole population. Afterward, environmental selection is used to prune population size. The specific changes are as follows:

A. AP-MODE

AP-MODE integrates current MODE with AP, based on evidence that combining recombining similar parents can boost its performance. The procedure and flowchart of the AP-MODE algorithm are depicted in Algorithm 4 and Fig. 1, respectively:

Algorithm 4 Main Steps of AP-MODE

Input: NP : population size; $G_{\max}$: Maximum Generation; β_0 : initial mating restriction probability.

Output: Non-dominated set of solutions

Step 1. Initialization population $P_t = \{x_1, x_2, ..., x_{NP}\}$ and generation counter $t = 1$

Step 2. Main loop:

2.1. Set external archive $E_t = \varnothing$

2.2. Custer the population: $Affinity\ Propagation(P_t)$ // Algorithm 3

2.3. For each $x_i \in P_t$, $i = 1, 2, ..., NP$ do

(a) identify B^i which denotes the set of solutions locating in the same cluster with x_i (except for x_i)

(b) set a mating pool Q^i for x_i , $Q^i = \begin{cases} B^i \setminus \{x_i\}, & \text{if } rand < \beta, \\ P^i \setminus \{x_i\}, & \text{otherwise,} \end{cases}$ rand is a random number in the range of $[0,1]$.

(c) generate a new solution $y_i = SolutionGeneration\left(Q^i, x_i\right)$ // Algorithm 5

(d) preserve the new solution $E_t = E_t \bigcup \{y_i\}$

2.4 Update the population $P_t = environmental\ selection(E_t \bigcup P_t)$ // Algorithm 6

2.5 Update the probability β based on non-dominate solutions

2.6 $t = t + 1$

Step 3. Stop iteration: if $t > NP$, output the result, otherwise turn to Step 2.

B. Solution generation

Since the differential evolution operator [40] usually outperforms other mutation operators in single-objective optimization, the MODE algorithm adopts it and the other common polynomial mutation (PM) [41] to improve performance, as shown in Algorithm 5.

Algorithm 5 Outline of Solution Generation

Input: parent solution x ; mating pool Q .

Output: candidate solution y ; external archive E

Step 1. Select two different random solutions x_{r1} and x_{r2} by the binary tournament selection approach.

Step 2. Create a candidate solution: $y' = x + F(x_{r1} - x_{r2})$

Step 3. Repair the candidate solution:

$$y_i'' = \begin{cases} a_i, y_i' < a_i \\ b_i, y_i' > b_i \\ y_i', \text{otherwise} \end{cases}$$

where a_i and b_i $(i = 1, 2, ..., n)$ are the lower and upper boundaries.

Step 4. Mutate the candidate solution

$$y_i''' = \begin{cases} y_i'' + \delta_i \times (b_i - a_i), & \text{if } rand() < pm, \\ y_i'', & \text{otherwise,} \end{cases}$$

$$\delta_i = \begin{cases} [2r + (1-2r)(\frac{b_i - y_i''}{b_i - a_i})^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} - 1, & \text{if } r < 0.5, \\ 1 - [2 - 2r + (2r-1)(\frac{y_i'' - a_i}{b_i - a_i})^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}}, & \text{otherwise,} \end{cases}$$

where $r = rand(), i = 1, 2, ..., n$

Step 5. Repair the solution

$$y_i = \begin{cases} a_i, & y_i''' < a_i, \\ b_i, & y_i''' > b_i, \\ y_i''', & \text{otherwise,} \end{cases} \quad \text{for } i = 1, 2, ...n$$

Step 6. Return $y = \{y_1, ..., y_n\}$ and update $E = E \bigcup \{y\}$

First, the binary tournament selection method randomly selects two solutions from the mating pool Q [42]. Then, the DE operator is utilized to produce a candidate by current individual x and two parents, x_{r1} and x_{r2} . Afterward, polynomial mutation (PM) mutates candidates to improve their quality. F and η_m denote the control parameter and the distribution index of mutation in these two operators, respectively. Sometimes, the newly generated candidates may violate the boundary constraints. In this case, we replace them with the closest boundary values. Furthermore, this method doesn't need the creation of a new candidate. It is worth mentioning that the DE's mutation operation is included only in AP-MODE, i.e., the selection operation is redefined in the next part. This design's purpose is that the DE mutation operator is not affected by any orthogonal coordinate rotation and can solve complex Pareto sets (PS) [43].

C. Environmental selection

Environmental selection aims to truncate the population to obtain the best individuals. In AP-MODE, the truncation, which consists of sorting the individuals with non-dominated sorting and evaluating individuals with crowding distance metric proposed in NSGA-II, is employed to establish a promising population. Algorithm 6 shows the procedure for environmental selection.

Algorithm 6 Outline of environmental selection

Input: current population $A_t \bigcup P_t$

Output: next generation population P_{t+1}

Step 1. Assign the solutions $x_i \in A_t \bigcup P_t$ to different fronts $F_1, ..., F_l, ...$ and calculate the crowding distance

Step 2. Add all individuals in P_t to P_{t+1}

Step 3. If $|P_{t+1}| > NP$ then

Step 4. Iteratively add individuals from F_i to P_{t+1} . If $|F_1| + \cdots + |F_{l-1}| = s < N$, $|F_1| + \cdots + |F_l| > N$, the next parent population P_t is constructed from the members of the sets $F_1, ..., F_{l-1}, ...$ and from $N - s$ members of the set F_l according to the PSA.

Step 5. End if

Step 6. Return P_{t+1}

The input is the current population $A_t \bigcup P_t$ of the size between NP and $2 \cdot NP$. Firstly, assign all individuals to different front $F_1, ...F_l, ...$ according to the non-dominated sorting approach. Next, calculate the crowding distance of the whole individual using the PSA method and sort all individuals in a descending order of each layer. Afterward, all the non-dominated solutions in $A_t \bigcup P_t$ are added to P_{t+1} . The new population is filled by fronts until $|P_{t+1}| > NP$. Iteratively add individuals from F_i to P_{t+1} . If $|F_1| + \cdots + |F_{l-1}| = s < N$, $|F_1| + \cdots + |F_l| > N$, the next parent population P_t is constructed from the members of the sets $F_1, ..., F_{l-1}, ...$ and from $N - s$ members of the set F_l according to the PSA. Fig. 2 illustrates the process of environmental selection.

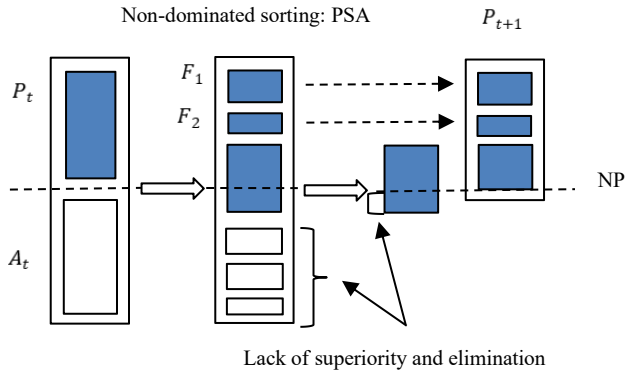


Fig. 2. An illustration of environmental selection

IV. COMPARISION

A. Experimental settings

In order to validate the performance of the proposed method, the experiments were conducted on AP-MODE, the original MODE, and NSGA-II. The experimental environment is 11th Gen Intel(R) Core(TM) i5-11320H @ 3.20GHz 3.19 GHz; 16 GB memory; Windows 11 operating system. The algorithm implementation platform is MATLAB R2015a.

1) Test Functions

To test the effectiveness of the proposed algorithm, a series of DTLZ [16] test functions proposed by Deb, Thiele et al. were adopted. Here, DTLZ1-DTLZ4 were used to test the performance of the above three algorithms in solving continuous or discontinuous distributions and preferences in different dimensions.

2) Algorithm Performance Evaluation Indexes

To evaluate the performance of AP-MODE, four metrics, including GD [45], IGD [46], HV [47], and Spacing [8], are employed to assess the convergence and diversity of the population of the above algorithms.

3) Parameter Settings

For comparison, all algorithms use the same initial population with the same genetic settings for each test. In NSGA-II, a polynomial mutation operator and a real-parameter simulated binary crossover operator are employed to form the genetic operators. The mutation and crossover probability are set to be $p_m = 1/n$ (n is the number of variables) and $p_c = 0.9$ respectively. In the AP-MODE and MODE, the mutation probability equals $p_m = 1/n$; the distribution index equals $\eta_m = 20$. Moreover, the initial mating restriction probability and DE control parameter are set to be $\beta = 0.5$ and $F = 0.5$, respectively. The maximum number of generations is $G_{\max} = 500$; the population size is N=100. In order to obtain significant results in this comparison, each algorithm is run independently 30 times.

B. Experimental Results Analysis

The mean and standard deviation values of four metrics are presented in TABLE I-IV. In these tables, the best index values are highlighted in bold. The results of a typical approximated set are showed in Fig. 3 and Fig. 4.

Now, we discuss the performance of AP-MODE. Fig. 3 (Fig. 4) presents the final approximated set for two (three) objectives of DTLZ1-4 with the median IGD values obtained by AP-MODE. Fig. 3 shows that the optimal solutions obtained by AP-MODE successfully converge to the Pareto front and completely cover all the instances in two objectives. In contrast, for DTLZ1 in two-dimensional objective space, the visual comparison shows that the solutions obtained through AP-MODE are unable to accurately approximate to the entire Pareto Fronts. For the two-dimensional objective space, we can observe that the objective points generated by AP-MODE distribute alone the PFs well from Fig. 4. Especially, as seen in Fig 4 (a), AP-MODE has a slight advantage in terms of diversity for DTLZ1 and the superiority is also manifested in the results for the DTLZ2. The most outstanding differences occur in the results for DTLZ3 and DTLZ4 on three objectives. According to [44], the quality of the final approximated set in DTLZ4 depends heavily on the initial population. AP-MODE consistently found solutions on the surface of the hyper-sphere using the same initial population.

The superiority of AP-MODE is also depicted in the statistics in TABLE I-IV. The mean metric values for each instance are presented and the numbers in the square brackets of TABLE I-IV show their ranks. The bold data in the table are the best mean metric values yielded by the algorithms for each instance.

The GD index represents the algorithm's convergence, while the IGD index reflects both the convergence and diversity of the approximated set. Smaller values of GD and IGD are preferred. As can be seen from TABLE I, AP-MODE obtains the best convergence performance among the three algorithms for the DTLZ1-4. Accordingly, the results show that the AP method outperforms the original MODE, whose framework embeds the AP approaches in terms of convergence. In a statistical sense, the best mean ranking of the average index values obtained by the three algorithms (called mean rank, MR) is AP-MODE, which reveals that it performs better than the competitors.

According to the results in TABLE II, our algorithm AP-MODE performs better than MODE in terms of convergence and distribution index. Compared with NSGA-II, AP-MODE has a narrow advantage for all the instances except DTLZ1. Statistical comparisons also show that AP-MODE performs best for the tests used in the experiments.

TABLE III shows the results of HV measurements. A higher HV value means better diffusion and convergence. According to TABLE IV, for DTLZ1-4, our algorithm (AP-MODE) obtains well results with the large HV values than MODE, but occasionally worse than NSGA-II. The results of the mean rank analysis indicate that AP-MODE performs similarly to NSGA-II. This could be due to the fact that while AP-MODE performs best with current parameter settings when dealing with DTLZ2 and DTLZ4 instances, it performs poorly with DTLZ1 and DTLZ3, resulting in a lower mean rank score for AP-MODE.

As we know, the spacing index indicates the diversity of the approximated set, with smaller values being preferable. Intuitively, as we can see from TABLE IV, the diversity values of AP-MODE are better than MODE and NSGA-II. The AP-MODE algorithm has excellent superiority in boosting diversity by embedding PSA. AP-MODE outperforms MODE in terms of solution convergence and

diversity. These observations demonstrate the effectiveness of the mechanism adopted in our AP based on the non-dominated counting approach and PSA to promote convergence and diversity.

V. CONCLUSION

The paper aims at the problems of low population diversity, slow convergence rate of the MODE algorithm in high-dimensional target spaces.To solve these problems, the AP-MODE is proposed in this paper. In AP-MODE, affinity propagation clustering is utilized to discover the structure of the Pareto solution set, whose structure is used to guide individual reorganization, thus conducting the search and accelerating convergence in the process of evolution. Then, the parent individuals for recombination originated from two sources. Moreover, according to the non-dominated counting approach, the mating restriction probability is adaptively updated at each generation. To demonstrate the potential benefit of this novel algorithm, a series of comprehensive experiments on benchmark instances are conducted. Based on the experiment's results, we can conclude that the AP-MODE algorithm has two distinct characteristics. Firstly, it assists in organizing individuals with similar traits. Secondly, it achieves convergence and exceptional diversity sets(based on IGD and HV).

In future work, certain strategies are suggested to decrease computational complexity while maintaining good performance and the proposed algorithm will be employed in practical optimization problems.

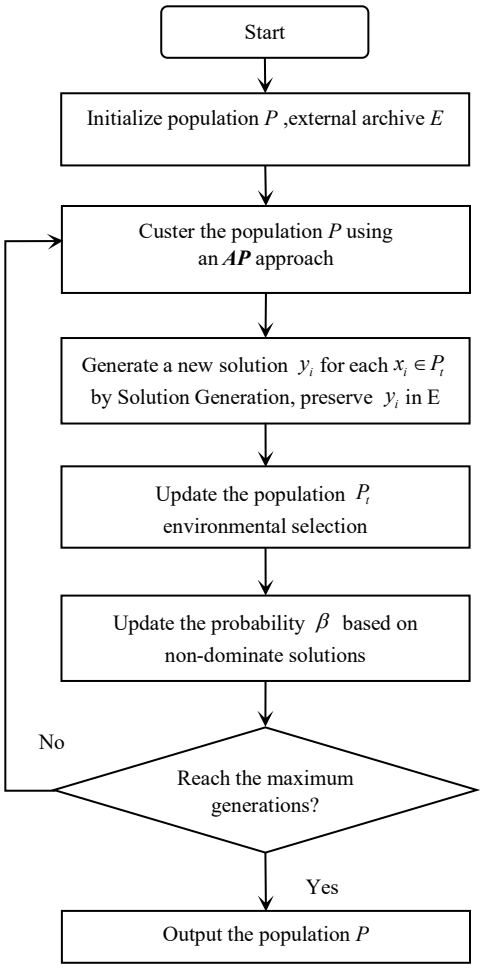
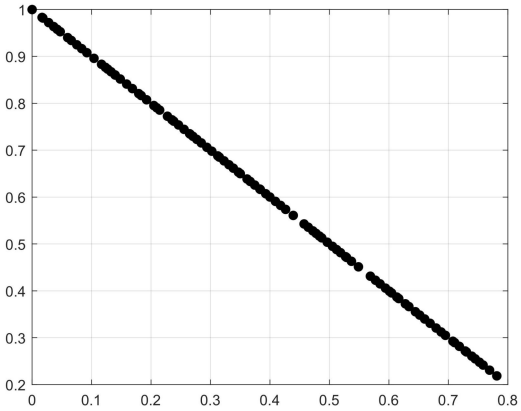
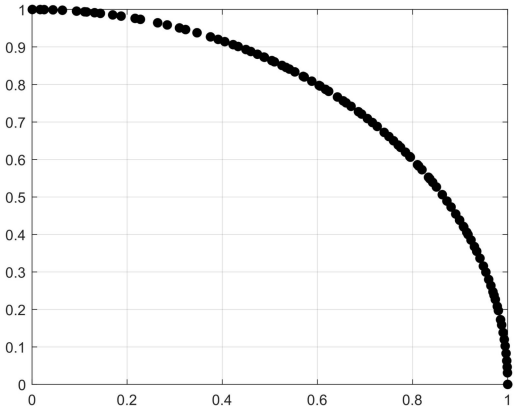


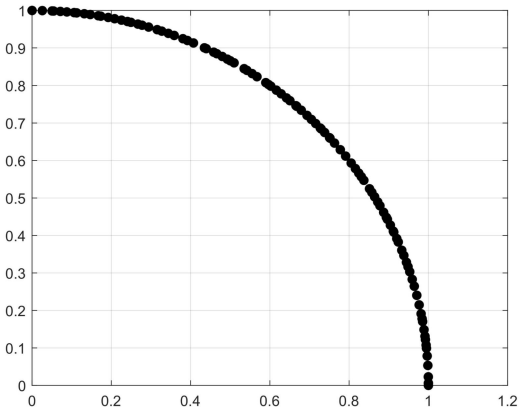
Fig.1. The flow chart of AP-MODE



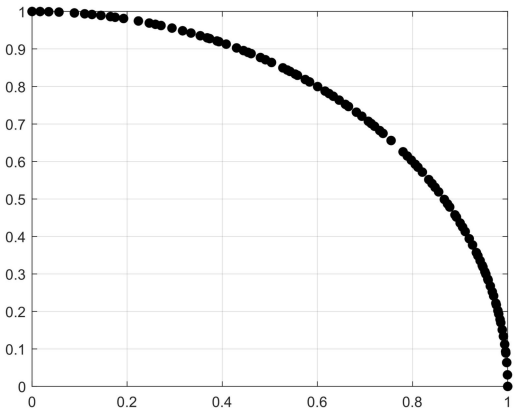
(a) DTLZ1



(b) DTLZ2

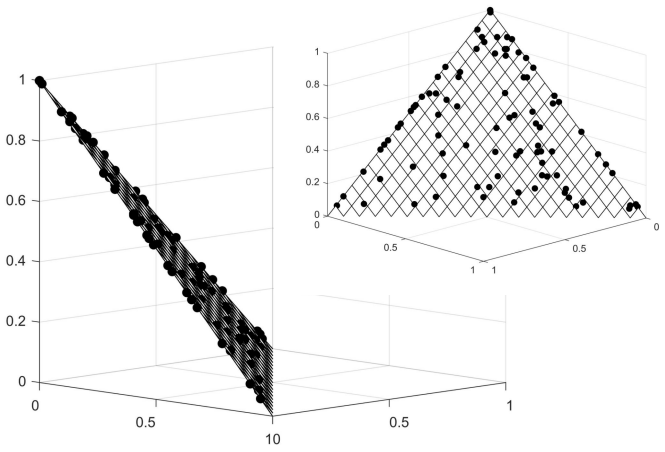


(c) DTLZ3

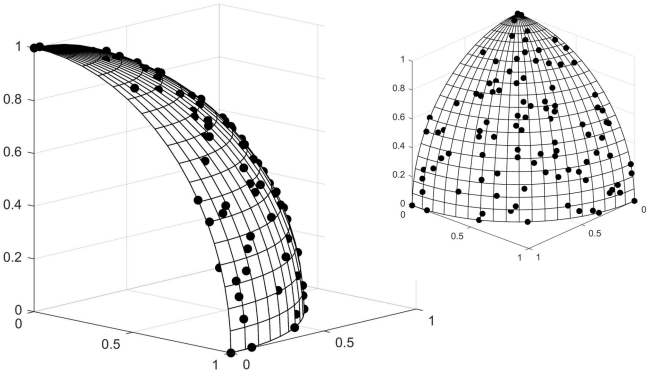


(d) DTLZ4

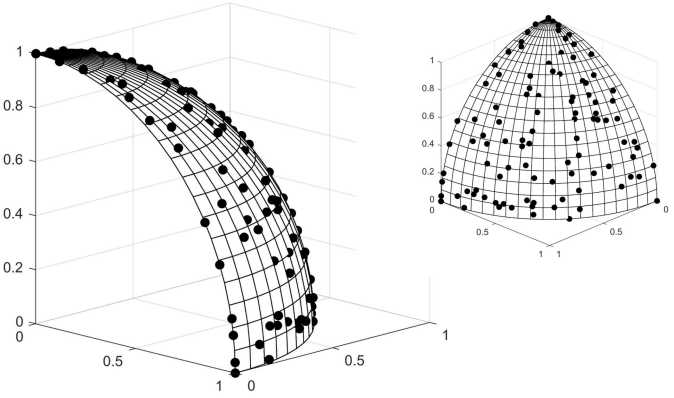
Fig. 3. Final solution sets on two objectives DTLZ1-4.



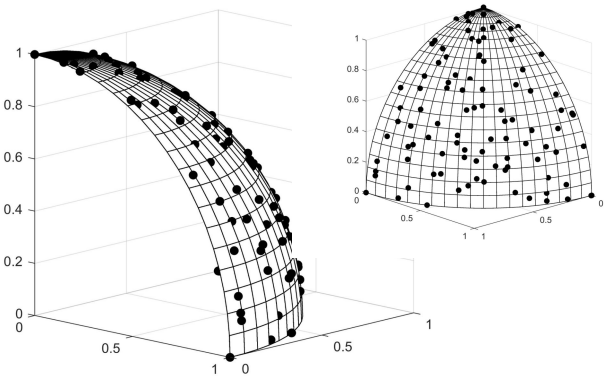
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



(d) DTLZ4

Fig. 4. Final approximated set on three objectives DTLZ1-4.

TABLE I. GD RESULTS OF FOUR APPROACHES

GD	NSGA-II	AP-MODE	MODE
DTLZ1	5.3290e-03 ± 4.2349e-04	**7.389724e-03 ± 2.42797e-04**	2.611807e-02 ± 2.60818e-03
DTLZ2	4.8263e-03 ± 2.2339e-04	**9.46727e-06 ± 2.48733e-06**	2.98286e-05 ± 1.01542e-05
DTLZ3	4.4104e-03 ± 5.8342e-04	**1.03079e-05 ± 1.83576e-06**	1.97311e-05 ± 1.90012e-06
DTLZ4	4.2920e-03 ± 5.2339e-04	**1.1474e-05 ± 2.10127e-06**	3.68182e-05 ± 2.49498e-05
Mean rank	0.3333	**0.2500**	0.4167

TABLE II. IGD RESULTS OF FOUR APPROACHES

IGD	NSGA-II	AP-MODE	MODE
DTLZ1	**4.1232e-03 ± 7.3396e-05**	4.35284 e-03 ± 2.41914e-05	5.20867e-03 ± 7.89105e-04
DTLZ2	4.5746 e-03 ± 4.3576e-04	**7.2334e-05 ± 1.90364e-05**	7.2672e-04 ± 2.76087e-05
DTLZ3	5.2629e-03 ± 3.3236e-04	**7.14497e-04 ± 1.32149e-05**	1.752846e-03 ± 2.47919e-04
DTLZ4	6.1483 e-03 ± 8.3676e-05	**7.19945e-04 ± 4.80988e-05**	4.478822e-03 ± 8.82224e-04
Mean rank	0.3333	**0.2500**	0.4167

TABLE III. HV RESULTS OF FOUR APPROACHES

HV	NSGA-II	AP-MODE	MODE
DTLZ1	**8.0113e-00 ± 2.3921e-03**	2.9908149 e-01 ± 2.827623e-03	1.531536e-01 ± 9.065461e-03
DTLZ2	7.3455e-01 ± 5.1771e-02	**7.4428122e-01 ± 3.841661e-03**	7.222772e-01 ± 5.48044e-03
DTLZ3	**7.2915e-00 ± 2.1301e-02**	7.4429027e-01 ± 2.83341e-03	7.075956e-01 ± 1.046824e-02
DTLZ4	7.1545e-01 ± 8.1201e-04	**7.4683908e-01 ± 5.80292e-03**	3.971449e-01 ± 6.3361959e-02
Mean rank	**0.2500**	**0.2500**	0.5000

TABLE IV. SPACING RESULTS OF FOUR APPROACHES

Spacing	NSGA-II	AP-MODE	MODE
DTLZ1	1.0718e-01 ± 1.2574e-03	**1.467099e-01 ± 4.9430706e-02**	1.19114e-01 ± 1.33865e-02
DTLZ2	1.1521e-01 ± 4.0920e-03	**1.395144e-01 ± 9.389004e-03**	1.371702e-01 ± 8.1863687e-03
DTLZ3	1.0927e-01 ± 1.9828e-03	**1.3584396e-01 ± 6.417196e-03**	1.334372e-01 ± 9.262767e-03
DTLZ4	1.1912e-01 ± 1.0433e-02	**4.117766e-01 ± 4.3166609e-03**	1.360292e-01 ± 8.441855e-03
Mean rank	0.5000	**0.1667**	0.3333

REFERENCES

[1] Y. C. Wan, S. D. Wang, Y. J. Hu, and Y. Y. Xie, "Multiobjective Optimization of the Storage Location Allocation of a Retail E-commerce Picking Zone in a Picker-to-parts Warehouse," Engineering Letters, vol. 31, no.2, pp. 481-493, 2023.

[2] Y. Hou, C. X. Wang, W. C. Dong, and L. X. Dang, "An Improved Particle Swarm Optimization Algorithm for the Distribution of Fresh Products," Engineering Letters, vol. 31, no.2, pp. 494-503, 2023

[3] N. Li, L. J. Wang, L. D. Lin, and H. J. Xuan, "Multi-objective Optimization Model and Improved Genetic Algorithm based on MOEA/D for VNF-SC Deployment," IAENG International Journal of Computer Science, vol. 50, no.1, pp. 172-181, 2023

[4] J. H. Li, "Research on Image Segmentation Technology and Intelligent Language Recognition Based on Ant Colony Algorithm," IAENG International Journal of Computer Science, vol. 50, no.2, pp. 553-563, 2023

[5] A. Wahba, R. El-khoribi, and S. Taie, "A New Hybrid Model for Energy Consumption Prediction Based on Grey Wolf Optimization," IAENG International Journal of Computer Science, vol. 49, no.2, pp. 469-481, 2022

[6] G. G. Chen, H. D. Zhuo, X. R. Hu, F. J. Long, and H. Y. Long, "Environment Economic Power Dispatch from Power System Based on Multi-objective Novel Tree Seed Optimization Algorithm," IAENG International Journal of Computer Science, vol. 48, no.4, pp. 845-861, 2021

[7] J. F. Zhao, J. Li, and C. R. Huang, "Multi-objective Optimization Model of Hydrodynamic Sliding Bearing Based on MOPSO with Linear Weighting Method," IAENG International Journal of Computer Science, vol. 48, no.3, pp. 738-745, 2021

[8] K. Deb, A. Pratap, S. Agrawal, T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," IEEE Trans. Evol. Comput., vol. 6, no. 3, pp. 182-197, 2002.

[9] K Deb and H Jain, "An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part i: Solving problems with box constraints", IEEE Transactions on Evolutionary Computation, vo. 18, no. 4, pp. 577-601, 2014.

[10] Q. Zhang, H. Li, "MOEA/D: A Multi-objective Evolutionary Algorithm Based on Decomposition,"IEEE Transactions on Evolutionary Computation., vol. 11, no. 3, pp. 712-731.

[11] K. Deb, M. Mohan, S Mishra, "Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of

Pareto-optimal solutions,"Evolutionary Computation., vol. 13, no. 4, pp. 501-525, 2005.
[12] E. Zitzler, M. Laumanns, L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm,"Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, 2001.
[13] K. V. Price, R. Storn, "Differential evolution-a simple evolution strategy for fast optimization," Dr. Dobb's Journal., vol. 22, pp. 18-24, 1997.
[14] X Li, M Yin, "Application of Differential Evolution Algorithm on Self-Potential Data," PLOS ONE., vol. 7, no. 12, 2012.
[15] X Li, K C Wong, "Elucidating Genome-Wide Protein-RNA Interactions using Differential Evolution,"IEEE/ACM Transactions on Computational Biology and Bio-informatics., vol. 16, no. 1, 2017.
[16] X. Li, Yin M, "Hybrid differential evolution and gravitation search algorithm for unconstrained optimization,"International Journal of Physical Sciences., vol. 6, no. 25, pp. 5961-5981, 2011.
[17] X. Li, M. Yin, "Differential Evolution for Prediction of Longitudinal Dispersion Coefficients in Natural Streams," Water Resources Management., vol. 27, no. 15, pp. 5245-5260, 2013.
[18] C. S. Chang, D. Y. Xu, H. B. Quek, "Pareto-optimal set based multi-objective tuning of fuzzy automatic train operation for mass transit system,"IEEE Proceedings-Electric Power Applications, vol. 146, no. 5, pp. 577-583, 1999
[19] H. A. Abbass, R. Sarker, C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems,"Congress on Evolutionary Computation. IEEE., vol. 2, pp. 971-978, 2001.
[20] F. Xue, A. C. Sanderson, R. J. Graves, "Pareto-based multi-objective differential evolution,"In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003), Canberra, Australia, IEEE Press., vol. 2, pp. 862-869, 2003.
[21] M Ali, P Siarry, M Pant, "An efficient Differential Evolution based algorithm for solving multi-objective optimization problems,"European Journal of Operational Research., vol. 217, no. 2, pp. 404-416, 2012.
[22] W. G. Li, S. J. Jia, C. H. Guo, "Decomposition-based Multi-objective Differential Evolution Algorithms and Their Applications,"Information and Control., vol. 42, no. 3, pp. 392-400, 2013.
[23] J. H. Zheng, L. Liu, M. Q. Li, et al, "Research on Differential Selection Strategy in Complex Multi-objective Optimization Problems,"Computer Research and Development., vol. 52, no. 9, pp. 2123-2134, 2015.
[24] Z. Liu, L. M. Zhen, "Multi-objective Differential Evolution Based on Jump Gene,"Computer Engineering., vol. 42, no. 4, pp. 168-172, 2016.
[25] X. Li, M. Yin, "Modified Differential Evolution with Self-adaptive Parameters Method,"Journal of Combinatorial Optimization., vol. 31, no. 2, pp. 546-576, 2016.
[26] B Ai, "Research on Multi-objective Differential Evolution,"Guilin University of Technology., Guilin, Guangxi, China, 2017.
[27] K. Li, H. X. Tian, "Adaptive Differential Evolution with Evolution Memory for Multi-objective Optimization,"IEEE Access., vol. 7, pp. 886-876, 2019.
[28] Y. Yang, J. Liu, S. Tan, & H. Wang, "A multi-objective diferential evolutionary algorithm for constrained multi-objective optimization problems with low feasible ratio," Appl. Sof Comput. 80, 42-56, 2019.
[29] Y. Tian, et al. "Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization" IEEE/CAA J. Autom. Sinica 9(10), 1801–1817, 2022.
[30] S. S. Rao, "Optimization theory and application,"New Delhi: Wiley Eastern Limited, 1991.
[31] J. G. Sun, J. Liu, L. Y. Zhao, "Research on Clustering Algorithms,"Journal of Software., vol. 19, no. 1, pp. 48-61, 2008.
[32] J. R. Chang, L. S. Chen, and L. W. Lin, "A Novel Cluster based Over-sampling Approach for Classifying Imbalanced Sentiment Data," IAENG International Journal of Computer Science, vol. 48, no.4, pp. 1118-1128, 2021.
[33] G. Wang, C. Xing, J. S. Wang, H. Y Wang, and J. X. Liu, "Clustering Validity Evaluation Functions of Fuzzy C-means Clustering Algorithm," IAENG International Journal of Computer Science, vol. 49, no.2, pp. 453-462, 2022
[34] B J Frey, D. Dueck, "Clustering by passing messages between data points,"Science., vol. 315, no. 5814, pp. 972-976, 2007.
[35] J. Meng, H. Hao, Y. Luan, "Classifier ensemble selection based on affinity propagation clustering,"J. Biomed. Inform., vol. 60, pp. 234-243, 2016.
[36] S. Salomon, G. Avigad, A. Goldvard, O. Schutze, "PSA-A New Scalable Space Partition Based Selection Algorithm for MOEAs," Eleventh Acm on Hypertext & Hypermedia., vol. 175, pp. 123-132, 2013.
[37] S. Salomon, et al, "PSA Based Multi Objective Evolutionary Algorithms,"In: Schuetze O. et al. (eds) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III. Studies in Computational Intelligence, vol 500. Springer, Heidelberg, pp. 233-259, 2014.
[38] D. Qu, X. F. Ding, H. M. Wang, "An Improved Multi-objective Algorithm: DNSGA2-PSA,"Journal of Robotics., vol. 2018, 2018.
[39] K. Li, S. Kwong, "A general framework for evolutionary multi-objective optimization via manifold learning," Neurocomputing., vol. 146, pp. 65-74, 2014.
[40] K Price, R. M. Storn, J. A. Lampinen, "Differential evolution: a practical approach to global optimization," Springer., New York, 2005.
[41] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms,"In: Proceedings of the 1st international conference on genetic algorithms., pp. 93-100, 1985.
[42] B. L. Miller, D. E. Goldberg, "Genetic Algorithms, Tournament Selection, and the Effects of Noise,"Complex Systems., vol. 9, no. 3, pp. 193-212, 1995.
[43] H. Li, Q. Zhang, "Multi-objective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," IEEE Trans. Evol. Comput., vol. 13, no. 2, pp. 284-302, 2009.
[44] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," Congress on Evolutionary Computation., vol.1, pp. 825-830, 2002.
[45] D. A. Van Veldhuizen, "Multi-objective evolutionary algorithms: Classifications, analyses and new innovations," Evolutionary Computation., vol. 8, no. 2, pp. 125-147, 1999.
[46] M. R. Sierra, C. A. C. Coello,"Improving PSO-based multi-objective optimization using crowding, mutation and -dominance," Evolutionary Multi-Criterion Optimization., vol. 3410, pp. 505 -519 , 2005.
[47] E. Zitzler,"Evolutionary algorithms for multi-objective optimization: Methods and applications," Ph.D. thesis, ETH, Zurich, Switzerland, 1999.
[48] W. Zheng and J. Scowcroft,"Quality of Service for Supporting Multimedia Applications," Selected Areas in Communications IEEE Journal on., vol. 14, no. 7, pp. 1228-1234, 1996.
[49] J. J. Wu, R. H. Hwang, and H. I. Lu,"Multicast Routing with Multiple QoS Constraints in ATM Networks," Information Sciences, vol. 124, no. 4, pp. 29-57, 2000.

BIBLIOGRAPHY

Dan Qu (1993-) received a Bachelor of Science degree in science from Mianyang Normal University, P.R. China, in 2016. She received a Master of Science degree at Southwest Petroleum University P.R. China in 2019.

She is a teaching assistant at the College of Mathematics and Statistics, Sichuan University of Science & Engineering and with the College of Mathematics Education, China West Normal University. Her research interests include multi-objective optimization, genetic algorithms, and so on.