A Many Objective Evolutionary Algorithm with Fast Clustering and Reference Point Redistribution

Mingde Zhao, *Hongwei Ge, Hongyan Han, Liang Sun
College of Computer Science and Technology
Dalian University of Technology
Dalian, China 116023
*Corresponding Email: hwge@dlut.edu.cn

***Abstract*—The design of effective evolutionary many-objective optimization algorithms is challenging for the difficulties to increase the evolution selection pressure for proximity as well as to maintain the diversity. In this paper, a fast Clustering based Algorithm with reference point Redistribution (fastCAR) is proposed. In the clustering process, a fast Pareto dominance based clustering mechanism is proposed to increase the evolution selection pressure that acts as a selection operator. In the redistribution process, a sampler is employed to gather information of the selection process and the reference lines are periodically redistributed by using a SVM classifier to maintain the diversity of the population. The experimental results show that the proposed algorithm fastCAR obtains competitive results on box-constrained CEC'2018 many-objective benchmark functions in comparison with 8 state-of-the-art algorithms.**

I. Introduction

Many real-world problems have more than one conflicting objectives to be optimized simultaneously [1], [2]. The multi-objective optimization problems can be mathematically described as follows:

Given the objective function to be optimized is $\boldsymbol{f}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})]^T$, where M is the number of objectives. Suppose $\boldsymbol{x}_1, \boldsymbol{x}_2$ are two solutions for the given problem, $\boldsymbol{x}_1$ **dominates** $\boldsymbol{x}_2$ *iff* $\forall i \in \{1, \cdots, M\}$, $f_i(\boldsymbol{x}_1) \leq f_i(\boldsymbol{x}_2)$ and $\exists j \in \{1, \cdots, M\}$, $f_j(\boldsymbol{x}_1) < f_j(\boldsymbol{x}_2)$. A solution $\boldsymbol{x}^*$ is **Pareto optimal** if no solution in the feasible solution space S dominates it. Due to the conflicting nature of this kind of problems, often there is not only one optimal solution. The collection of all Pareto optimal solutions is called the Pareto Set (PS). In normal occasions, the size of PS is usually infinitive. The mappings of solutions in the PS to the objective space O via the objective function $\boldsymbol{f}(\cdot)$ constitute the true Pareto Front (PF), which acts as the limit of the evolutions of all objectives.

The goal is to find a finite solution set X^* such that using the limited individuals, the mappings of individuals from X^* to O via $\boldsymbol{f}(\cdot)$ could be both an proximate and a diverse representative of the true PF. To conclude, the crucial task of multi-objective optimization evolutionary algorithms (MOEAs) is *obtaining proximity* while *maintaining diversity*.

Traditional MOEAs face challenges when there are more than 3 objectives [2], [3]. This is mainly caused by the two following factors: (1) Deterioration for Pareto dominance relation: the Pareto dominance based sorting is both costly and inefficient for many objectives. The insufficient pressure provided on evolution of the population cannot ensure the proximity to the true PF. This phenomenon is known as the "dominance resistance" [4]; (2) The aggravation of the conflict between convergence and diversity [3]. To address the two problems, different kinds of solutions emerge.

To deal with the lack of evolution pressure, modification on the original Pareto dominance relation is being researched. By adopting the relaxed dominance relations such as ϵ-dominance [5], α-dominance [6], etc., sufficient pressure for evolution selection towards the true PF can be obtained [7]. Nevertheless, the adopted relaxing relations are often parameter sensitive, as the degree of relaxation has a key impact on the evolution [8].

To deal with the conflict between proximity and diversity, algorithms based on decomposition such as MOEA/D [9] and NSGA-III [10] have been contributed. Treating reference lines as weight vectors or space segmentation references, an original MOP is decomposed into a number of simpler optimization problems to be optimized collaboratively. Though obtaining better performance, the performance of this kind of algorithm is sensitive to the distribution of the reference lines, the leading factor of maintaining diversity. Thus decomposition based algorithm with niching and reference line adaptation mechanisms such as A-NSGA-III [11] are also being researched. Indicator based algorithms such as IBEA [12] have also been contributed since they are less haunted by the problems of the reference lines. However, though achieving competitive performance, the calculation of the indicator incurs intensive burden on computation. Comprehensive methods are also investigated. A representative algorithm, ARMOEA [2], combines the IGD-NS [13] based selection, which is indicator based and capable of distinguishing solutions that have no contribution, as well as a decomposition based reference point adaptation method that could provide better versatility.

Based on the problems and motivations reviewed, in this paper, the studies are conducted along the following outlines:

1) To address the deterioration of Pareto dominance as the number of objectives increases, a selection operator, the *fast clustering* has been contributed. In the selection process, a fast selection for the non-dominated solutions of the first front and a two-level intraclass ranking scheme are proposed. These provide sufficient evolution

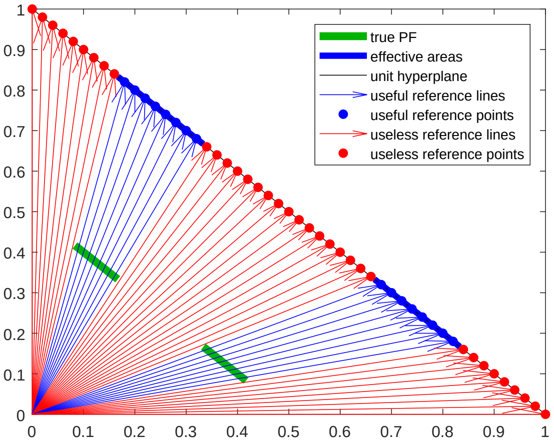


Fig. 1. Demonstration for the central projection of the true PF covering only parts of the unit hyperplane

selection pressure on the population towards the true PF and maintain the diversity using reference lines.

2) To address the impact on the performance for the sensitivity to reference lines of the decomposition based algorithms, a learning classifier based potential active *reference points redistribution* mechanism is proposed to reduce the number of useless reference lines and generate appropriate number of reference points in the effective areas on the unit hyperplane so that appropriate number of reference lines are evenly going through the true PF and providing better guidance for evolution towards diversity.

II. Proposed Algorithm

In this section, the proposed algorithm fastCAR is explained with focus on the two crucial processes, the first of which is the selection operator of fast clustering mechanism employed to guide the evolution of the solutions using the reference lines, while the second of which is the redistribution process that employs a classifier to redistribute reference lines.

A. Fast Clustering

As the first step, due to the inconsistency of the magnitude of the objectives, normalization is applied on all solutions. For convenience, "solution" here refers to the mapping of a solution from the solution space S to the objective space O via the objective function $\boldsymbol{f}(\cdot)$. The normalization process is identical to the normalization method in [10].

1) Frontier Solution Selection: Instead of using the traditional costly non-dominated sorting based on Pareto dominance relation for all solutions in every front, a *frontier solution selection* mechanism is employed, which only picks out the non-dominated solutions in the first front of the current population. For convenience, these solutions are called the *frontier solutions*, which are the the most desirable solutions in the current population.

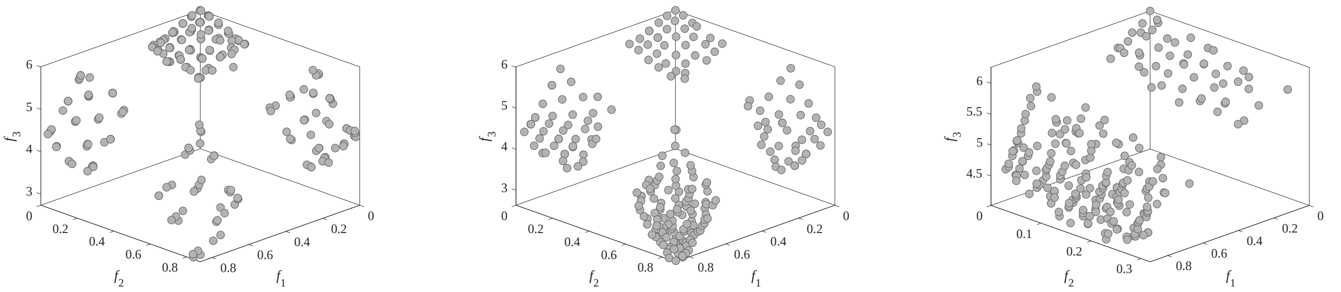
2) Cluster Generation: Each frontier solution is assigned to its nearest reference line by calculating DI, the sine values of the included angle of the solution and all reference lines. Reference lines with one or more frontier solutions are called *active reference lines*. The frontier solutions belonging to the same active reference line are sorted using the proposed proximity and diversity indicator (PDI).

$$PDI(\boldsymbol{o},\boldsymbol{r}) = PI(\boldsymbol{o})+DI(\boldsymbol{o},\boldsymbol{r}) = mean(\boldsymbol{o})+sin(\boldsymbol{o},\boldsymbol{r}) \quad (1)$$

$DI(\boldsymbol{o},\boldsymbol{r}) = sin(\boldsymbol{o},\boldsymbol{r})$ is the calculated sine value of the angle between the frontier solution $\boldsymbol{o}$ and the assigned active reference line $\boldsymbol{r}$, representing the distribution error between the reference line $\boldsymbol{r}$ and the frontier solution $\boldsymbol{o}$. The smaller the overall DI, the closer the solutions are to the reference lines in the objective space. So, by employing a certain mechanism that distributes the reference lines to intersect with the true PF at the evenly distributed points, DI directs the evolution of solutions toward a better diversity. $PI(\boldsymbol{o})$ is an indicator calculating the mean value of all objectives, representing the proximity for solution $\boldsymbol{o}$. The smaller the $PI(\boldsymbol{o})$, the closer the frontier solution is to the true PF. Similar to PBI in [9], PDI is also a metric to evaluate a non-dominated solution using a reference line. The proximity term for PBI guides the evolution of solutions towards the origin, while PDI guides the evolution of solutions in a direction perpendicular to the unit hyperplane. The two directions both have the similar effect of pushing the solutions towards the true PF. However, the evolution direction of the proximity term of PDI is more even for the irregular PFs with long tails or sharp heads as the guiding directions are parallel towards the unit hyperplane. Also, the calculation of PDI is faster, that only a mean value of all objectives of each solution needs to be additionally calculated.

For each active reference line, the attached frontier solutions are gathered as a cluster. Also, the frontier solution with the best PDI is taken as the center of the cluster. All non-frontier solutions are assigned to the clusters with the nearest centers. For each cluster, the non-frontier solutions are sorted by their Euclidean distances to the corresponding cluster centers. In this way, the mappings of the nearest frontier solutions are used to guide the evolution of the non-frontier solutions. By employing a criterion that does not select the solutions outside the first front as guiding solutions, a more intense pressure on evolution selection is obtained towards better proximity. After the aforementioned operations, clusters are created with two intraclass sorted queues: the sorted frontier solution queue and the sorted non-frontier solution queue.

3) Round-robin Selection: To enhance diversity by increasing the chance of selecting non-frontier solutions into the next generation, a round-robin selection method is employed. In each cluster, the selection queue is obtained by concatenating the sorted non-frontier solution queue after the sorted frontier solution queue. In each round, in a shuffled sequence, the head of each selection queue is popped out as a candidate for the next generation until the number of individuals for the next generation reaches N.



(a) Case I: less reference lines than needed

(b) Case II: appropriate number of reference lines

(c) Case III: more reference lines than needed

Fig. 2. Demonstration for the impact of the number of reference lines in the effective area

B. Reference Line Redistribution

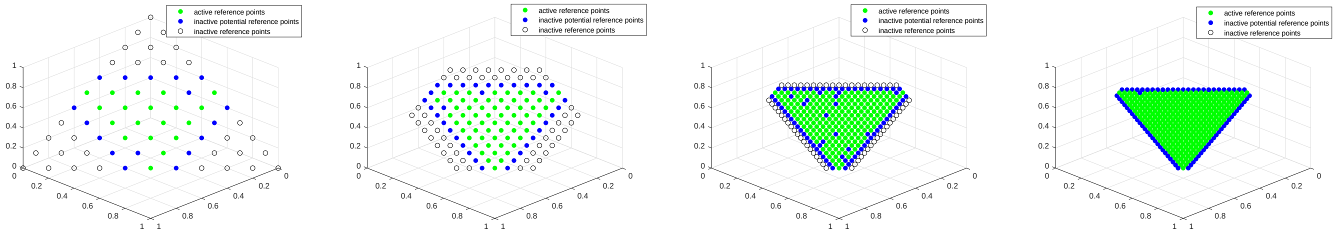
Traditionally, reference points are evenly generated in a unit hyperplane in the objective space with the unit intercept on each objective with the expectation of intersecting the true PF evenly. While the central projection of the true PF (the effective areas for reference generation) from the origin to the hyperplane may cover just a part of the reference points (for convenience, the PF of this kind is called the part-covering PF). The other points are useless for evolution and the calculation on their corresponding reference lines is meaningless. Assuming that the true PF is part-covering, when solutions evolve to be nearer the true PF, the reference lines that intersect with the true PF are more likely to be active than the ones that do not intersect the true PF. When N reference points are uniformly generated on the unit hyperplane, the goal of fast clustering is to create clusters with one solution assigned to one reference line. However this cannot be done since less than N reference lines are actually useful. Thus a viable solution is to redistribute the useless reference lines so that more of them can intersect with the true PF, *i. e.*, to move the reference points into the effective area of the unit hyperplane. An example is given in Fig. 1. When the reference points are evenly generated on the unit hyperplane, the reference lines are generated as lines that go through the origin and the reference points. However, as the example problems has a disconnected true PF, only a part of reference lines can intersect with the true PF, as shown in Fig. 1 where the reference lines intersecting with the true PF are in blue while others are in red. Thus by a central projection from the origin, the true PF is projected on the unit hyperplane as the effective areas for the reference points. As it is shown in Fig. 1, the blue reference points are generated inside the central projection of the true PF from the origin on the unit hyperplane, while the red ones are not. On one hand, if this is the case, that the density of reference lines is low and less than N reference lines go through the true PF, there will be less than N reference lines guiding the evolution of population. There will always be a cluster with more solutions than it is expected to attach. On the other hand, if the density of the reference points is too dense and the useful areas of the unit hyperplane are overcrowded with more reference points than needed, the solutions may unevenly attach to only a part of reference lines that go through only a part of true PF. Our aim is to appropriately redistribute the reference points in the useful areas. The three situations are shown in Fig. 2. In Fig. 2 (a), where insufficient reference points are distributed in the effective area of the unit hyperplane, it can be observed that some appropriately distributed reference line are overcrowded with more than 1 solutions. In Fig. 2 (b), where the reference points are appropriately distributed, it can be observed that the distribution of the solutions are relatively even in the true PF, which is the ideal case. In Fig. 2 (c), where too many reference points are generated, it can be observed that only a random part of reference lines are attached with solutions, causing an incomplete distribution of solutions near the true PF.

Algorithm 1: Redistribution

```
Input: active reference points $ARP$, inactive reference
       points $IRP$, sampler status $S$, old reference point
       generation density $D$, number of objectives $M$,
       population size $N$
Output: reference points $RP$, new generation density $D$
if isStable(S) and $|ARP| < N$ then
    //construct training data;
    TrainingData = $\{\langle ARP, 1\rangle, \langle IRP, 0\rangle\}$;
    [SVM, $\delta$] = newGaussianSVM(TrainingData);
    //increase reference generation density;
    $D = D + 1$;
    points = generateReferencePoints(density, M);
    //pick the points whose Platt scaling scores $p >= \delta$;
    $RP$ = SVMPick(SVM, points, $\delta$);
```

With the tense pressure on evolution selection provided by fast clustering, the population approaches the true PF quickly. Thus, the reference points corresponding to the active reference lines are more likely to be in the useful areas on the hyperplane. Though the characteristics of the true PF is not a known *priori*, using the status of all reference lines, the useful areas can be gradually estimated.

This task is carried out by employing a status sampler and a fine-grained Gaussian SVM. Whenever the distribution of



(a) (b) (c) (d)

Fig. 3. Demonstration for the impact of the number of reference lines in the effective area: (a) Initial stage: N reference points are uniformly generated in the unit hyperplane; (b) Reference points are more densely generated and the low-scored ones are reduced using the scoring of the classifier whose knowledge is learnt from (a); (c) The process repeats: the number of active reference points gradually increases and the number of inactive reference points decreases; (d) The final status for reference point distribution: sufficient reference points are generated in the effective area.

active reference lines are stable, which means that in a period of cycles the active reference lines do not change, the sampler records and analyzes the attributes (active or inactive) of all reference lines. If only a part of reference lines are active and the number of active reference lines are smaller than N, which means insufficient useful reference lines are generated, the reference point corresponding to the reference lines will be marked with their attributes and be fed to SVM for learning. Then the generation density for reference points is increased to generate more reference points and the old ones will be discarded. By employing Platt scaling [14], the output of the SVM predicting whether a certain reference pointed is located in the effective area or not can be converted in to p, a score of probability distribution over classes. If $p > 0.5$, the reference point is considered as useful and vice versa. However, the learning may be inaccurate for many reasons. Thus, a threshold δ, $\delta < 0.5$, is employed to pick out the additional potential reference points whose $\delta \leq p < 0.5$. In our implementation, threshold δ is dynamically set to be the score of reference point with the N-th highest score, *i. e.*, when the number of active reference lines is below N, after a denser generation of the reference points, at least N reference points will be kept. Thus, at least N reference points are continuously redistributed to the current potential areas unless the disabling criteria is met.

Take a run of fastCAR on the benchmark problem MaF1 for demonstration. At first, N reference points are uniformly generated on the unit hyperplane and their corresponding reference lines are handed to fast clustering for population selection as it is shown in Fig. 3 (a). Then, the selection operator additionally returns the attributes of the reference lines. The active reference points corresponding to the active reference lines (the reference lines with frontier solutions attached) are marked in green, while the other reference points are inactive. If the sampler considers this status stable, the attributes of all current reference points will be sent to the classifier for learning. After the learning is complete, using the Platt scaling scoring scheme, the SVM classifier is able to identify the potential of a reference point by scoring it. In Fig. 3 (a)(b)(c)(d), the blue points are the potential inactive reference points located in the potentially effective area of the unit hyperplane with scores greater or equal to δ. Considering that there are not enough active reference lines, more reference points are generated using a higher generation density. By applying the classifier on the newly generated reference points, the points with lower score than threshold, the ones not likely to be active, are reduced. Only the points with scores higher than or equal to δ are kept for the next clustering process. In Fig. 3 (b), as it is demonstrated, the reference points located at the tails of the triangle area are reduced for their low scores, only the reference points with scores greater than or equal to δ are participating the clustering process. Whenever the sampler considers the current status of reference lines stable and the number of current active reference points is below N, the above process repeats, as it is demonstrated in Fig. 3 (c)(d). As the process repeats, more accurate boundaries for classification can be obtained, since the population is getting nearer to the true PF and the intermediate areas on the unit hyperplane are filled with denser data points. If the fast clustering process reports that the number of active reference lines is almost or above N, the reference point redistribution process including the sampler, the learning classifier and the reference point generator is disabled. The pseudo code for the redistribution process is in Algorithm 1. The advantage is that we do not have to use a definitive model for the boundaries of the projection of the true PF, a SVM is used to gradually estimate the boundary of the effective area with no extra burden on the fitness evaluation of the population, as the information of learning comes from a periodic sampler that gathers the information in each stable clustering process. This also ensures the versatility on the MaOPs with disconnected PFs.

C. Proposed Framework

The framework of the proposed algorithm is formulated as the flowchart in Fig. 4.

First, to initialize, N individual solutions are randomly generated in the solution space S and N reference points are uniformly generated in the unit hyperplane in the objective space O representing the direction for N reference lines. Then, the main loop cycles until the criteria of termination is satisfied: evolve to get the offsprings using the N solutions in the population and obtain the potential population with

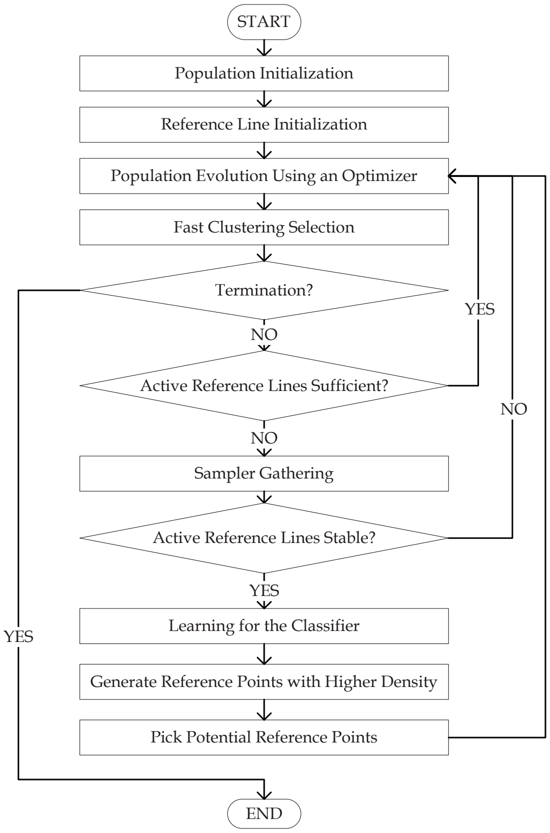

START
Population Initialization
Reference Line Initialization
Population Evolution Using an Optimizer
Fast Clustering Selection
Termination?
YES
NO
Active Reference Lines Sufficient?
NO
NO
Sampler Gathering
Active Reference Lines Stable?
YES
YES
Learning for the Classifier
Generate Reference Points with Higher Density
Pick Potential Reference Points
END

Fig. 4. The Framework of the Proposed Algorithm

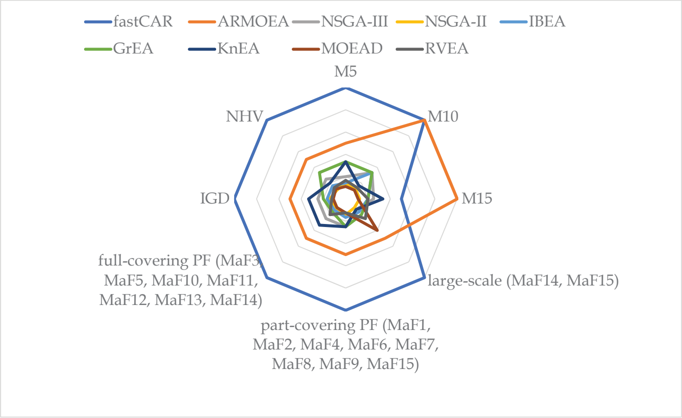


Fig. 5. The radar diagram for the specified performance

a size of $2N$. Then, out of the potential population, using the selection operator of fast clustering, the new population with the size of N is picked out. Then the information of the clusters is additionally used. If the number of active reference lines are not enough, a SVM will learn from the stable status and locate the potential areas for the newly denser generated reference points.

III. Experimental Studies

A. *Experimental Settings*

To demonstrate the effectiveness of fastCAR, the benchmark suite for CEC'2018 MaOP competition is chosen, which is designed for understanding the strengths and weaknesses of multi and many objective evolutionary algorithms with benchmark functions of diverse characteristics. In the technical report for CEC'2018 MaOP competition [3], 15 benchmark functions are provided with box constrains in the solution space. And all the benchmarks are running with $M = 5$, $M = 10$ and $M = 15$. With official standard parameter settings including using the default "EAreal" optimizer with default parameters and the population size of $N = 240$ and the designated number of fitness evaluations, all data of each algorithm given in this section is averaged over 20 independent runs on PlatEMO. The details of the benchmark functions and the standard for CEC'2018 MaOP competition can be found in [3]. The source code of fastCAR is implemented using the open source PlatEMO [15] with MATLAB R2017b[1].

B. *Performance Metrics*

The two prevailing metrics to evaluate the quality of a solution population are IGD and HV.

Originally used in [16], IGD, short for Inverted Generational Distance, is a comprehensive measure of both proximity and diversity of a population. Let P^* be a set of evenly distributed sample points over the true PF and P be the current population, the n-order IGD from P to P^* can be formulated as:

$$IGD(P, P^*, n) = \frac{(\sum_{\boldsymbol{v}\in P^*} d^n(\boldsymbol{v}, P))^{1/n}}{|P^*|} \tag{2}$$

where $d(\boldsymbol{v}, P)$ is the minimum Euclidean distance between a sample point $\boldsymbol{v}$ on the true PF and the solutions in P. Originally, when IGD is first used in evaluating the performance of MOEAs [17], [18], n is taken as 2, *i. e.*, the 2-order IGD is adopted. In the recently papers, n is widely chosen as 1 as it is in [2]. In this paper, to concur with the metrics of the CEC'2018 MaOP competition [3], n is also taken as 1. When $|P^*|$ is large enough, $IGD(P, P^*)$ can measure both the diversity and the proximity of P. The smaller the IGD, the better the proximity and diversity.

Calculating the volume of the objective space between the obtained solution population and a reference point, HV, short for HyperVolume, can give the set a comprehensive assessment in terms of proximity and diversity [19]. HV gives the volume enclosed by the union of the polytopes $p_1, p_2, \cdots, p_k$, where each p_i is formed by the intersections of the following hyperplanes arising out of $\boldsymbol{x}_i$, along with the axes: for each axis in the objective space, there exists a hyperplane perpendicular to the axis and passing through the point $\boldsymbol{f}(\boldsymbol{x}_i)$ [7]. For clarity and the consistency with the CEC'2018 MaOP competition, the normalized HV (NHV) [3] is adopted to normalize the hypervolume to the interval $[0, 1]$.

[1] The MATLAB source code can be found in Mingde Zhao's github: https://github.com/PwnerHarry/

TABLE I
IGD RESULTS OF DIFFERENT ALGORITHMS AVERAGED FROM 20 INDEPENDENT RUNS

		fastCAR	ARMOEA	NSGA-III	NSGA-II	IBEA	GrEA	KnEA	MOEAD	RVEA
MaF1	5	**1.12E-01**	1.40E-01	1.97E-01	1.73E-01	1.45E-01	1.35E-01	1.24E-01	2.34E-01	3.07E-01
	10	2.96E-01	2.53E-01	2.67E-01	2.92E+00	2.74E-01	2.49E-01	**2.31E-01**	4.68E-01	6.76E-01
	15	3.24E-01	2.79E-01	3.17E-01	3.37E-01	3.16E-01	3.38E-01	**2.76E-01**	5.77E-01	6.89E-01
MaF2	5	**1.08E-01**	1.11E-01	1.29E-01	1.43E-01	1.09E-01	1.13E-01	1.29E-01	1.24E-01	1.29E-01
	10	2.65E-01	2.03E-01	2.41E-01	1.63E-01	1.90E-01	4.02E-01	**1.56E-01**	3.24E-01	2.36E-01
	15	2.57E-01	2.25E-01	2.08E-01	**1.75E-01**	2.96E-01	3.69E-01	1.79E-01	3.63E-01	6.84E-01
MaF3	5	**6.43E-02**	9.66E-02	1.08E-01	2.86E-01	5.83E-01	7.13E-01	1.53E-01	1.25E-01	8.37E-02
	10	**8.31E-02**	8.60E-02	2.66E+02	2.37E+06	3.46E-01	4.59E+02	1.67E+08	1.45E-01	8.51E-02
	15	**8.97E-02**	9.17E-02	1.88E-01	2.47E+06	2.65E-01	6.52E+02	1.28E+09	1.31E-01	1.21E-01
MaF4	5	**2.01E+00**	2.65E+00	3.21E+00	2.23E+00	5.81E+00	2.20E+00	3.07E+00	9.66E+00	4.88E+00
	10	7.81E+01	1.02E+02	9.73E+01	**5.94E+01**	1.65E+02	1.61E+02	7.59E+01	4.45E+02	2.34E+02
	15	3.24E+03	4.69E+03	4.17E+03	**1.62E+03**	4.51E+03	5.00E+03	1.80E+03	1.90E+04	7.23E+03
MaF5	5	**1.97E+00**	2.21E+00	2.21E+00	2.29E+00	2.34E+00	2.13E+00	2.48E+00	9.23E+00	2.94E+00
	10	8.82E+01	9.63E+01	8.81E+01	9.53E+01	5.88E+01	**4.82E+01**	7.63E+01	3.01E+02	1.03E+02
	15	2.43E+03	3.63E+03	2.59E+03	1.72E+03	1.32E+03	**1.09E+03**	1.71E+03	7.32E+03	2.94E+03
MaF6	5	6.51E-03	**4.21E-03**	4.40E-02	4.66E-03	6.07E-02	3.75E-02	8.09E-03	9.59E-02	8.15E-02
	10	**2.49E-02**	1.19E-01	2.16E-01	3.85E-01	1.65E-01	3.93E-01	9.78E+00	1.23E-01	1.37E-01
	15	**6.56E-02**	2.47E-01	3.17E-01	6.20E-01	3.92E-01	1.29E+00	2.53E+01	1.85E-01	1.20E-01
MaF7	5	2.91E-01	3.28E-01	3.28E-01	3.59E-01	3.07E-01	2.65E-01	**2.90E-01**	1.08E+00	4.49E-01
	10	1.60E+00	1.69E+00	1.02E+00	1.82E+00	9.04E-01	2.49E+00	**8.67E-01**	2.00E+00	2.96E+00
	15	1.48E+01	3.50E+00	4.74E+00	3.26E+00	**2.47E+00**	6.82E+00	2.54E+00	4.09E+00	3.95E+00
MaF8	5	**9.86E-02**	1.24E-01	2.35E-01	1.52E-01	9.20E-01	1.51E-01	2.59E-01	2.95E-01	4.56E-01
	10	2.98E-01	**1.34E-01**	3.37E-01	1.56E-01	1.02E+00	1.47E-01	1.66E-01	9.40E-01	8.00E-01
	15	6.05E-01	1.72E-01	4.02E-01	1.54E-01	1.03E+00	1.56E-01	**1.37E-01**	1.37E+00	1.21E+00
MaF9	5	**9.34E-02**	1.20E-01	4.36E-01	5.74E-01	9.95E-01	1.45E+00	8.25E-01	1.42E-01	4.61E-01
	10	2.32E-01	**1.76E-01**	7.10E-01	7.11E+01	1.05E+00	1.40E+00	2.46E+01	1.06E+00	8.41E-01
	15	6.40E-01	**1.48E-01**	3.75E-01	8.82E-01	3.17E+00	5.86E+00	3.71E-01	3.09E+00	1.23E+00
MaF10	5	5.02E-01	4.75E-01	**4.57E-01**	9.34E-01	5.56E-01	6.13E-01	5.21E-01	1.22E+00	4.67E-01
	10	**5.02E-01**	1.24E+00	1.07E+00	1.75E+00	1.09E+00	1.25E+00	1.22E+00	2.62E+00	1.27E+00
	15	1.63E+00	1.90E+00	1.55E+00	2.02E+00	**1.49E+00**	2.15E+00	1.62E+00	3.13E+00	1.83E+00
MaF11	5	**6.58E-01**	8.35E-01	8.25E-01	8.92E-01	1.30E+00	1.24E+00	6.75E-01	5.13E+00	1.70E+00
	10	4.76E+00	2.45E+00	4.97E+00	**2.01E+00**	5.81E+00	2.77E+00	2.22E+00	1.67E+01	6.93E+00
	15	6.21E+00	**3.17E-01**	1.04E+01	8.80E-01	9.99E+00	5.73E+00	5.81E+00	2.73E+01	1.80E+01
MaF12	5	**9.35E-01**	1.12E+00	1.12E+00	1.32E+00	1.16E+00	1.09E+00	1.17E+00	1.56E+00	1.12E+00
	10	4.60E+00	4.67E+00	4.59E+00	4.92E+00	**4.11E+00**	4.20E+00	4.57E+00	8.89E+00	4.52E+00
	15	7.80E+00	7.59E+00	7.92E+00	8.23E+00	6.70E+00	6.86E+00	**6.53E+00**	1.50E+01	7.17E+00
MaF13	5	1.30E-01	**1.29E-01**	2.59E-01	1.71E-01	7.65E-01	3.51E-01	2.24E-01	1.74E-01	6.85E-01
	10	3.23E-01	**1.20E-01**	2.18E-01	1.29E-01	1.11E+00	3.18E-01	1.45E-01	9.97E-01	8.95E-01
	15	4.43E-01	1.42E-01	3.10E-01	**1.06E-01**	1.47E+00	5.81E-01	1.12E-01	1.47E+00	1.11E+00
MaF14	5	**3.52E-01**	3.63E-01	1.49E+00	1.01E+00	1.65E+00	7.33E-01	5.46E-01	7.67E-01	7.92E-01
	10	6.95E-01	6.04E-01	2.54E+00	6.74E+00	1.08E+00	2.00E+00	9.66E+00	**5.55E-01**	6.36E-01
	15	7.28E-01	**5.93E-01**	1.33E+00	3.71E+00	1.16E+00	7.57E+00	1.11E+01	9.46E-01	7.72E-01
MaF15	5	**4.21E-01**	6.24E-01	1.18E+00	1.97E+01	8.57E-01	1.01E+00	4.15E+00	5.80E-01	5.88E-01
	10	9.97E-01	**8.47E-01**	1.24E+00	9.37E+01	7.65E-01	7.60E-01	7.02E+00	9.77E-01	9.76E-01
	15	1.14E+00	1.30E+00	3.37E+00	1.07E+02	1.31E+00	1.30E+00	1.03E+02	**1.07E+00**	1.18E+00

The higher the NHV, the better the size of the space covered, the better proximity and diversity.

C. Results for CEC'2018 MaOP Competition

The IGD and NHV data are compared with the results of the highlighted state-of-the-art algorithms in the PlatEMO platform including ARMOEA [2], NSGA-III [10], NSGA-II [20], IBEA [12], GrEA [21], KnEA [22], MOEAD [9] and RVEA [1]. The IGD and NHV results on CEC'2018 MaOP benchmark of fastCAR and the 8 compared algorithms are presented in Table I and Table II, respectively.

Friedman tests are also respectively conducted on the IGD results and the NHV results, the rankings and the detailed results are presented in Table III. The Friedman test shows that the rankings are effective for the differences of the results among the 9 algorithms are statistically significant on both two metrics with $\alpha = 0.05$. With the significant statistical differences, the proposed algorithm fastCAR achieves the highest performance by obtaining the top ranking in both IGD and NHV.

D. Categorized Analysis

With $M = 5$, the proposed fastCAR outperforms other algorithms by ranking the 1st in 11 benchmark functions out of 15 in both IGD and NHV. When dealing with the 8 functions with part-covering PF (MaF1, MaF2, MaF4, MaF6, MaF7, MaF8, MaF9 and MaF15), where the central projection of the true PF covers only a part of the hyperplane, the proposed algorithm obtains the second best in IGD and the best in NHV. Though achieving top performance with $M = 10$ and competitive performance with $M = 15$, the deterioration for the performance can be observed. This may be resulted by the

TABLE II
NHV results of different algorithms averaged from 20 independent runs

		fastCAR	ARMOEA	NSGA-III	NSGA-II	IBEA	GrEA	KnEA	MOEAD	RVEA
MaF1	5	**1.20E-02**	8.96E-03	5.31E-03	6.32E-03	9.78E-03	9.94E-03	1.06E-02	5.58E-03	2.29E-03
	10	4.67E-07	0.00E+00	**4.89E-07**	0.00E+00	4.41E-07	6.22E-07	4.11E-07	3.51E-08	4.95E-09
	15	**8.07E-12**	0.00E+00	6.04E-12	0.00E+00	7.04E-12	0.00E+00	0.00E+00	9.16E-14	3.23E-14
MaF2	5	1.86E-01	1.66E-01	1.55E-01	1.52E-01	1.87E-01	**1.95E-01**	1.89E-01	1.62E-01	1.52E-01
	10	2.08E-01	2.14E-01	2.16E-01	1.87E-01	2.17E-01	**2.43E-01**	1.69E-01	2.09E-01	1.86E-01
	15	1.18E-01	1.63E-01	1.43E-01	1.08E-01	**2.17E-01**	2.15E-01	1.10E-01	1.81E-01	5.30E-02
MaF3	5	**9.99E-01**	9.99E-01	9.94E-01	8.99E-01	5.58E-01	3.77E-01	9.68E-01	9.84E-01	9.96E-01
	10	**1.00E+00**	9.99E-01	0.00E+00	0.00E+00	8.81E-01	0.00E+00	0.00E+00	9.67E-01	9.99E-01
	15	**1.00E+00**	9.99E-01	9.53E-01	0.00E+00	9.55E-01	0.00E+00	0.00E+00	9.67E-01	9.97E-01
MaF4	5	**9.79E-02**	7.16E-02	6.39E-02	7.86E-02	6.76E-03	1.19E-01	1.08E-01	1.66E-02	9.35E-03
	10	3.13E-06	2.98E-06	**2.05E-04**	3.04E-05	2.51E-08	4.56E-04	1.16E-04	3.17E-08	6.23E-08
	15	1.10E-11	4.98E-11	1.88E-07	0.00E+00	6.07E-14	**5.63E-07**	6.83E-09	9.87E-14	2.18E-13
MaF5	5	**8.12E-01**	7.58E-01	7.80E-01	6.51E-01	7.76E-01	7.71E-01	7.75E-01	4.41E-01	7.61E-01
	10	**9.68E-01**	9.65E-01	9.68E-01	0.00E+00	9.73E-01	9.70E-01	9.54E-01	4.33E-01	9.54E-01
	15	**9.90E-01**	9.74E-01	9.90E-01	0.00E+00	9.96E-01	9.93E-01	9.94E-01	3.10E-01	9.44E-01
MaF6	5	**1.29E-01**	1.29E-01	1.26E-01	1.29E-01	1.14E-01	1.18E-01	1.28E-01	1.23E-01	1.14E-01
	10	**9.91E-02**	8.71E-02	8.24E-02	2.84E-02	9.28E-02	1.99E-02	0.00E+00	8.54E-02	8.97E-02
	15	**9.24E-02**	6.88E-02	6.50E-02	8.27E-03	4.46E-02	2.50E-03	0.00E+00	4.00E-02	9.20E-02
MaF7	5	2.57E-01	2.37E-01	2.37E-01	1.91E-01	**2.63E-01**	2.59E-01	2.56E-01	8.19E-03	2.15E-01
	10	1.61E-01	1.52E-01	1.77E-01	2.82E-06	**1.97E-01**	2.18E-01	8.57E-02	3.17E-04	1.50E-01
	15	1.91E-02	8.20E-02	1.06E-01	9.55E-10	1.62E-01	**1.88E-01**	5.37E-06	6.19E-08	1.10E-01
MaF8	5	**1.22E-01**	1.16E-01	9.22E-02	1.06E-01	5.67E-03	1.11E-01	1.02E-01	1.04E-01	7.65E-02
	10	6.80E-03	**1.06E-02**	8.81E-03	8.89E-03	6.92E-05	9.78E-03	1.01E-02	6.52E-03	4.97E-03
	15	5.66E-05	6.30E-04	5.08E-04	5.18E-04	2.62E-07	5.61E-04	**6.24E-04**	3.17E-04	1.57E-04
MaF9	5	**3.16E-01**	3.01E-01	1.92E-01	1.63E-01	4.78E-02	3.83E-02	1.29E-01	2.85E-01	1.69E-01
	10	1.35E-02	**1.57E-02**	7.21E-03	2.33E-05	8.14E-04	2.62E-03	3.88E-04	1.12E-02	4.25E-03
	15	2.67E-04	**1.24E-03**	8.21E-04	3.13E-04	2.40E-05	7.18E-05	8.93E-04	6.14E-04	2.52E-04
MaF10	5	9.44E-01	**9.98E-01**	9.98E-01	9.65E-01	9.86E-01	9.71E-01	9.88E-01	9.40E-01	9.96E-01
	10	9.98E-01	**9.99E-01**	9.83E-01	9.96E-01	9.93E-01	9.85E-01	9.98E-01	5.39E-01	9.97E-01
	15	9.98E-01	9.99E-01	1.00E+00	**1.00E+00**	9.96E-01	9.93E-01	9.98E-01	5.87E-01	9.99E-01
MaF11	5	9.95E-01	**9.96E-01**	9.96E-01	9.81E-01	9.75E-01	9.71E-01	9.91E-01	9.53E-01	9.89E-01
	10	**9.96E-01**	9.96E-01	9.97E-01	9.97E-01	9.86E-01	9.80E-01	9.94E-01	9.37E-01	9.88E-01
	15	9.95E-01	9.97E-01	9.94E-01	**9.98E-01**	9.91E-01	9.73E-01	9.94E-01	9.42E-01	9.72E-01
MaF12	5	**7.76E-01**	7.18E-01	7.15E-01	5.53E-01	7.34E-01	7.48E-01	7.49E-01	6.05E-01	7.40E-01
	10	8.98E-01	8.45E-01	8.95E-01	6.22E-01	8.88E-01	9.00E-01	**9.13E-01**	3.35E-01	8.89E-01
	15	8.26E-01	8.93E-01	8.53E-01	6.35E-01	9.12E-01	**9.19E-01**	9.35E-01	2.69E-01	8.36E-01
MaF13	5	**2.61E-01**	2.51E-01	2.00E-01	2.24E-01	2.54E-03	1.53E-01	2.06E-01	2.31E-01	1.53E-01
	10	**1.32E-01**	1.30E-01	1.16E-01	1.21E-01	2.30E-05	6.99E-02	1.12E-01	5.49E-02	7.91E-02
	15	5.24E-02	**9.10E-02**	4.03E-02	8.51E-02	6.33E-09	3.93E-04	8.39E-02	3.48E-02	5.64E-02
MaF14	5	**6.01E-01**	5.38E-01	1.17E-01	1.63E-02	6.75E-03	3.05E-01	4.23E-01	1.87E-01	2.05E-01
	10	5.24E-01	5.67E-01	4.50E-02	0.00E+00	8.73E-02	8.09E-02	0.00E+00	**7.54E-01**	6.65E-01
	15	4.39E-01	**6.72E-01**	0.00E+00	1.62E-02	2.00E-02	5.72E-02	0.00E+00	1.71E-01	3.90E-01
MaF15	5	**4.28E-02**	1.90E-02	7.07E-06	0.00E+00	1.25E-03	1.04E-03	2.84E-08	3.86E-02	2.34E-02
	10	4.43E-07	4.72E-06	0.00E+00	0.00E+00	2.92E-05	**3.00E-05**	0.00E+00	2.92E-06	1.81E-06
	15	5.14E-11	0.00E+00	0.00E+00	0.00E+00	6.13E-16	1.07E-15	0.00E+00	**2.19E-10**	1.85E-13

TABLE III
Results of Friedman Test on IGD and NHV for the Compared Algorithms

Friedman-IGD				Friedman-NHV			
	Rank	χ^2	$1-p$		Rank	χ^2	$1-p$
fastCAR	**3.29**	61.84	99.99%	**fastCAR**	**6.73**	54.79	99.99%
ARMOEA	3.41			ARMOEA	6.42		
NSGA-III	5.00			NSGA-III	5.19		
NSGA-II	5.49			NSGA-II	3.58		
IBEA	5.37			IBEA	4.50		
GrEA	5.21			GrEA	5.30		
KnEA	4.49			KnEA	4.94		
MOEAD	6.81			MOEAD	3.84		
RVEA	5.93			RVEA	4.49		

sparse distribution of the reference points which is insufficient for the classifier to obtain the correct boundaries for the effective area of the unit hyperplane. On other benchmark problems (MaF3, MaF10, MaF11, MaF12, MaF13 and MaF14) with whole-covering PF projection area, the proposed algorithm obtains leading performance on both IGD and NHV when $M = 5$, $M = 10$ or $M = 15$. On the large-scale problems (MaF14 and MaF15), where the dimensionality of the solution space is relatively high, the proposed algorithm also achieves the highest performance with respect to the two metrics.

Fig. 5 presents the radar diagram specifying the ranking of the performance on each type of function of the compared algorithms. In the radar diagram, all rankings in the dimensions are acquired by Friedman test ranking. For the category M5, M10 and M15, all 15 functions are averaged in both IGD and NHV. The ranking of the corresponding dimension of the radar diagram is obtained by averaging the IGD and NHV rankings. For the functions of different characteristics, the data of $M = 5$, $M = 10$ and $M = 15$ are all taken to Friedman tests with both IGD and NHV. The rankings of dimensions of IGD and NHV are directly taken from the previous overall Friedman tests.

IV. Conclusion

To address the problems in the existing MOEAs, this paper proposes a MOEA with fast clustering mechanism and reference point redistribution (fastCAR). In the proposed algorithm, a fast clustering mechanism maintains the balance between proximity and diversity and addresses the problems of insufficient pressure of evolution towards the true PF. To make the reference lines more properly distributed, fastCAR is also integrated with the redistribution process for reference points, so that the diversity of the population is maintained and the problem for the sensitivity of decomposition on reference lines is addressed. Experimental results show that fastCAR obtains competitive results on the CEC'2018 many-objective benchmark functions compared with the state-of-the-art MOEAs. It can be concluded that the proposed algorithm fastCAR is effective.

Acknowledgment

The authors are grateful to the support of the National Natural Science Foundation of China (61572104, 61103146, 61402076), the Fundamental Research Funds for the Central Universities (DUT17JC04), and the Project of the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University (93K172017K03).

References

[1] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, Oct 2016.

[2] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2017.

[3] R. Cheng, M. Li, Y. Tian, X. Xiang, X. Zhang, S. Yang, Y. Jin, and X. Yao, "Benchmark functions for cec'2018 competition on many-objective optimization," CERCIA, School of Computer Science, University of Birmingham Edgbaston, Birmingham B15 2TT, U.K., School of Computer Science and Technology, Anhui University Hefei 230039, China, School of Computer Science and Informatics, De Montfort University Leicester, LE1 9BH, U.K., Department of Computer Science, University of Surrey Guildford, Surrey, GU2 7XH, U.K., Tech. Rep., 2017.

[4] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, Dec 2007.

[5] K. Deb, M. Mohan, and 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.

[6] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of pareto-based moeas: does non-dominated really mean near to optimal?" in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 2, 2001, pp. 957–962 vol. 2.

[7] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artificial Intelligence*, vol. 228, no. Supplement C, pp. 45 – 65, 2015.

[8] M. Laumanns and R. Zenklusen, "Stochastic convergence of random search methods to fixed size pareto front approximations," *European Journal of Operational Research*, vol. 213, no. 2, pp. 414 – 421, 2011.

[9] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Dec 2007.

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

[11] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, Aug 2014.

[12] E. Zitzler and S. Künzli, *Indicator-Based Selection in Multiobjective Search*. Springer Berlin Heidelberg, 2004, pp. 832–842.

[13] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, "A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, July 2016, pp. 5222–5229.

[14] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[15] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, Nov 2017.

[16] P. Czyżżak and A. Jaszkiewicz, "Pareto simulated annealing: a metaheuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.

[17] C. A. Coello Coello and M. Reyes Sierra, *A Study of the Parallelization of a Coevolutionary Multi-objective Evolutionary Algorithm*. Springer Berlin Heidelberg, 2004, pp. 688–697.

[18] M. Reyes-Sierra and C. Coello, "A new multi-objective particle swarm optimizer with improved selection and diversity mechanisms," Technical Report EVOCINV-05-2004, Sección de Computación, Depto. de Ingeniería Eléctrica, CINVESTAV-IPN, México, Tech. Rep., 2004.

[19] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, Nov 1999.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[21] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, Oct 2013.

[22] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, Dec 2015.