A Geometric Structure Based Particle Swarm Optimization Algorithm for Multi-objective Problems

Wenqiang YUAN, Yusheng LIU, *Member, IEEE*, Hongwei WANG, Yanlong Cao

Abstract— This article presents a novel evolutionary strategy for multi-objective optimization in which a population's evolution is guided by exploiting the geometric structure of its Pareto front. Specifically, the Pareto front of a particle population is regarded as a set of scattered points on which interpolation is performed using a geometric curve/surface model to construct a geometric parameter space. On this basis, the normal direction of this space can be obtained and the solutions located exactly in this direction are chosen as the guiding points. Then, the dominated solutions are processed by using a local optimization technique with the help of these guiding points. Particle populations can thus evolve towards optimal solutions with the guidance of such a geometric structure. The strategy is employed to develop a fast and robust algorithm based on correlation analysis for solving the optimization problems with more than three objectives. A number of computational experiments have been conducted to compare the algorithm to another three popular multi-objective algorithms. As demonstrated in the experiments, the proposed algorithm achieves remarkable performance in terms of the solutions obtained, robustness and speed of convergence.

Index Terms—Geometric structure, Non-uniform rational B-spline, Evolution strategy, Multi-objective optimization

I. INTRODUCTION

ENGINEERING optimization problems generally involve multiple objectives. Searching for an optimal solution to meet multiple objectives at the system level is a complicated task as the objectives often conflict with each other. A lot of research has been undertaken to develop effective methods for Multiple-Objective Optimization (MOO) problems, which predominantly falls into two categories. Specifically, the first category focuses on converting all the objective functions of a MOO problem to a single function or choosing one of them as the main objective while rearranging the others as constraints, i.e. transforming it into a problem with only one objective. The other one employs a different approach from reducing the number of objective functions, emphasizing finding a Pareto front or a representative subset of the functions that do not dominate other functions.

Additionally, the Evolutionary Algorithms (EAs) have also been adapted to solve MOO problems by using customized fitness functions and solution selection strategies to improve diversity of solution and speed of convergence. For example,

Wenqiang Yuan and Yusheng Liu are with Zhejiang University, College of Computer Science and Technology, Hangzhou 310058, China (email: wqyuan@zju.edu.cn; yslu@cad.zju.edu.cn)

Hongwei WANG is with School of Engineering, University of Portsmouth, Portsmouth, UK, PO1 3DJ (hongwei.wang@port.ac.uk).

Yanlong Cao is with Zhejiang University, College of Mechanical Engineering, Hangzhou 310058, China (email: sdcaoyl@zju.edu.cn)

many variations of the Genetic Algorithm (GA) rank the individuals in a population using a specific fitness function and select good individuals according to their fitness values for producing a new generation, i.e. preserving some excellent genes. The elitism Non-dominated Sorting Genetic Algorithm [1] and the Pareto-archived EA [2] are among the pioneering work of evolutionary MOO. These methods have achieved significant advantages over classical optimization methods particularly in attaining fast convergence speed and good diversity of solutions.

Particle Swarm Optimization (PSO) [3] is a bio-inspired meta-heuristic based method predominantly for solving single objective problems. It has become increasingly popular due to good generality and outstanding performance. Some variants on PSO have been proposed and developed to solve MOO problems, which can obtain an excellent Pareto front with a good diversity of solutions. Traditional PSO-based methods update the speed of each particle through calculating its local best position and the global best position. Beausoleil [4] proposed a Multiple Objective Scatter Search (MOSS) algorithm using a Tabu/scatter hybrid searching method for solving MOO problems. Although the Multi-Objective Particle Swarm Optimization (MOPSO) methods have been proved to be able to achieve good performance, they still have several inadequacies. Firstly, there are no strategies employed to ensure that a uniform distribution of solutions can be attained. Secondly, the best feasible flight direction is very unpredictable, leading to instability and thus affecting speed of convergence. Thirdly, it is difficult to make approximation of the points outside the feasible region. Last but not least, their effectiveness of solving engineering design problems is questionable as in these problems sometimes an extreme value can be expected in one aspect even though this may lead to higher costs in other aspects. Thus a big research gap has been raised by these inadequacies.

This research aims to address this gap by developing a Geometric-Structure-based PSO (GSPSO) method. Its main idea is that the inherent geometric structure of the current generation's Pareto front is exploited to guide particles flying directly. The advantages of GSPSO are two folds. Firstly, distribution of points in the current Pareto front can be understood clearly in the geometric parameter space. Thus the ideal regions are very predictable as the normal direction indicates better solutions within the feasible space. Secondly, uniformity of the current Pareto front can be easily achieved by performing interpolation based on the geometric structure of the current Pareto front. Even the solutions with extreme

values can also be obtained in a similar way. Therefore, the inadequacies mentioned above can be addressed by GSPSO.

The rest of this paper is organized as follows. Section II introduces the background of this study and discusses related work. Section III outlines the proposed method and Section IV details the dimension reduction method. Section V introduces the optimization process based on the guiding points and Section VI describes the uniformity operation. Section VII details a number of computational experiments conducted for evaluating the GSPSO method and comparing it with some popular MOO algorithms. Finally, Section VIII highlights the main conclusions of this work and discusses possible future work.

II. RELATED WORK

A. Particle Swarm Optimization

During a PSO procedure, particles in a swarm fly through a hypercube search space. They keep changing their positions inside the search space in response to the social interactions with other individuals to emulate each other's behavior. The scope of a particle's neighborhood depends on the specific network topology selected for a swarm, determining the information exchanging routes of all the particles.

Let $\bar{x}_i(t)$ be the position of particle i at the t -th iteration. $\bar{x}_i(t)$ is then updated during each time of iteration by adding a velocity $\bar{v}_i(t)$ to it. This position updating operation can be formally represented using the following formula:

$$\bar{x}_i(t+1) = \bar{x}_i(t) + \bar{v}_i(t+1) \quad (1)$$

The velocity vector is used to describe the information exchanged during social interactions and can be calculated as follows:

$$\bar{v}_i(t+1) = W\bar{v}_i(t) + C_1r_1(\bar{x}_{pbest_i} - \bar{x}_i(t)) + C_2r_2(\bar{x}_{gbest} - \bar{x}_i(t)) \quad (2)$$

In Equation (2), C_1 is the cognitive learning factor which controls the influence of the personal best particles effect; C_2 is the social learning factor which controls the influence of the global best particles; W is the inertia weight employed to control the influence of the previous history of velocity values on the current velocity value for each particle; $\bar{x}_{pbest_i}$ is the personal best position of the particle i ; $\bar{x}_{gbest}$ is the global best position of all the particles in the entire population; and r_1 and r_2 are random values within the range of interval $[0, 1]$.

B. PSO for Multi-Objective Optimization

PSO has a number of advantages over other evolutionary optimization algorithms particularly in its relatively simple implementation. Also, PSO methods have good capability of solving complex problems, fast speed of convergence and good distribution of solutions. Therefore, it has been applied to many engineering applications and so far several effective PSO variants have been proposed [5]-[7]. In particular, the Optimized MOPSO (OMOPSO) [8] is a classic variant which creates an external archive based on the crowding distance from NSGA-II to filter out leader solutions and uses mutation operators to accelerate the convergence of the swarm. In recent years, some approaches have been proposed to extend

the original PSO algorithm to deal with MOO problems, such as the sigma method [9], the dominated tree [10] and the dynamic multiple swarms [11]. These approaches differ from each other predominantly in the different leader-selection strategies employed.

Research has also been done to propose more effective evolutionary strategies, leading to several new PSO variants. For example, an adaptive strength Pareto PSO was used to handle MOO problems and three hybrid EA-PSO algorithms were developed for different problems. In this method, all the particles are involved in exchanging information to direct the swarm towards good points in the search space and particle positions are updated by referring to the good points reserved in a particular archive. In addition, an improved Developed Multi-Objective PSO (D²MOPSO) that incorporated both dominance and decomposition was also proposed in [12]. In D²MOPSO, decomposition aims to simplify a MOO problem by transforming it into a set of aggregation problems. A multi-objective PSO based on two local best solutions was proposed in [13] which emphasized that the global best or local best should be chosen from the top fronts in a non-domination sorted external archive with a reasonably large size. The PSO implementation with a parallel computing strategy was also explored to improve the efficiency of the algorithm [14].

There are a handful of PSO variants adopting the geometric information of a population. For example, John and Cameron applied a method based on the Non-Uniform Rational Basis Spline (NURBS) meta-models for the robust optimization of mixed-integer problems [15]. This method allows for a new and powerful definition of robustness for integer variables in classic problems. However, this work mainly concentrated on the robustness of optimization problems while the geometric information was not used to improve the evolution strategy.

C. The NURBS Model

NURBS is a mathematical model that is commonly used in computer aided design and computer graphics to generate and represent curves and surfaces [16]. It offers great flexibility and precision for handling both analytic and freeform shapes and thus is part of several industry-level standards such as the Standard for the Exchange of Product model data (STEP) [17].

A NURBS curve is a recursively-defined spline curve and its standard formula is given as follows:

$$S(u) = \frac{\sum_{i=1}^n b_i w_i N_{i,k}(u)}{N \sum_{i,k} N_{i,k}(u)} \quad \text{for } a \leq u \leq b \quad (3)$$

In Equation (3), b_i is a vector that defines the location of the i -th control point (the total number is n_c); w_i is a positive scalar weight associated with the i -th control point; $N_{i,k}(u)$ is the B-spline basic function given as a parametric function of u ; and k is the order of the curve or surface. Parameter u defines a position along the curve that is equivalent to a point on the curve defined by the vector $S(u)$. The B-spline basic function is a recursive function that is defined using the following equations:

$$N_{i,k}(u) = \left(\frac{u - x_i}{x_{i+k-1} - x_i} \right) N_{i,k-1}(u) + \left(\frac{x_{i+k} - u}{x_{i+k} - x_{i+1}} \right) N_{i+1,k-1}(u) \quad (4)$$

$$N_{i,0}(u) = \begin{cases} 1 & x_i \leq u \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In Equations (4) and (5), x is the knot vector, which is a monotonic sequence of parameter values that define the region of influence for the control points within a NURBS meta-model. For the i -th control point, its region of influence is defined by the meta-model order k .

Similarly, the NURBS surface is defined in a 3D space using the following parametric equation:

$$S(u, v) = \frac{\sum_{i=1}^n \sum_{j=1}^m b_{i,j} w_{i,j} N_{i,p}(u) N_{j,q}(v)}{\sum_{i=1}^n \sum_{j=1}^m b_{i,j} N_{i,p}(u) N_{j,q}(v)} \quad (6)$$

for $a_u \leq u \leq b_u$ & $a_v \leq v \leq b_v$

Fig. 1. NURBS curve and NURBS surface.

III. OVERVIEW OF THE GSPSO METHOD

Fig. 2. Overview of the GSPSO process

The main idea of GSPSO is to regard the Pareto front of the current population as a set of geometric points that can be

fitted to establish a NURBS equation. This equation can be used to acquire the geometric structure information of the current Pareto front and such information can be used to direct the flight for particles. It should be noted that a NURBS curve/surface can always be generated for the given Pareto front points since the current Pareto front cannot be closed like a sphere. The entire process of using the GSPSO method is illustrated using the flowchart in Fig. 2, which can be divided into three main steps.

The number of objectives of an optimization problem needs to be checked firstly. If the number is more than three, a dimension reduction process needs to be conducted based on the correlation coefficient matrix as the NURBS curve and surface models can only deal with problems with two and three variables.

Secondly, the evolution operation is performed on each generated population by exploiting the geometric structure of the current Pareto front. This operation involves establishing the parametric NURBS equation to define the current Pareto front and on this basis obtaining the guiding points to control the particles' flight.

Finally, some operations on the Pareto front are conducted to ensure of uniformity and extensibility based on the trust region method [18].

IV. DIMENSION REDUCTION

Quite often optimization objectives conflict with each other or have very low correlation. In this sense, any two objectives can be called positively correlated objectives, negatively correlated objectives or non-correlated objectives depending on the degree correlation between them. By analyzing this degree of correlation, dimension reduction can be conducted

for optimization problems and the GSPSO method can thus be used for the optimization problems with more than three objectives. In this study, a dimension reduction method is proposed based on correlation analysis. The main reason for using such a method is that it is straightforward and easy to implement, although some dimension reduction methods have been developed elsewhere such as those discussed in [22-24]. Additionally, orthogonal design experiments are conducted to obtain the most appropriate solution.

A. Dimension Reduction for Multiple Objectives Based on Correlation

Correlation is an indicator that shows the degree of linear correlation between two random variables. First proposed by Pearson in 1880, this term has been widely used in many disciplines [25].

For a multi-objective optimization problem, let the number of objectives be d and the size of the sample population for each objective be n . After the sample population is generated, the space of objective functions can be regarded as a matrix of size $n*d$. Specifically, each column of the matrix means a set of sample values of one objective while each row represents an individual in the optimization problem.

In order to explore the whole search space of each design variable and reduce the complexity of experiments to a reasonable extent, the orthogonal experiment design method [26] is adopted in this study. The primary characteristics of this method include being homogeneous and comparable. For a specific problem, a number of experiments need be conducted according to a chosen orthogonal table. After all the experiments are completed, some clusters of solutions are obtained which represent different states. In this work, the cluster of solutions with the most common clustering results is selected since the dependency relationships within such a cluster are quite representative.

To better describe the dimension reduction method based on correlation analysis, two definitions are given as follows.

Definition 1: *Objective point cluster* is a set composed of one or more objective points.

Definition 2: *Objective point cluster distance:* suppose that the objective point cluster C_1 is composed of m objective points and C_2 is composed of n objective points. In addition, let r_{ij} be the correlation coefficient between the i -th objective point in C_1 and the j -th objective point in C_2 . The cluster distance cd is then defined as follows:

$$cd = \frac{\sum_{i=1}^m \sum_{j=1}^n r_{ij}}{mn} \quad (7)$$

Further to the analysis mentioned above, the dimension reduction process for the objective space based on an unsupervised clustering analysis method can be described using the following two steps:

- 1) Calculate the cluster distance cd for any two objective point clusters (each objective point is initially considered as one objective point cluster);
- 2) Merge the two objective point clusters with the largest cd into one and decrease the total number of objective point clusters by one;

These two steps will be repeated until the number of objective point clusters is equal to a given value (three or two in this study).

B. Analysis of the Fuzzy Correlation Semantic Information

After the dimension reduction process is completed, each objective point cluster contains at least one objective. Ideally, any pair of objectives in an objective point is positively correlated after dimension reduction. As shown in Case C_1 in Fig. 3, the points A_1 , B_1 and D_1 are all positively correlated. However, it is actually quite common for an objective point cluster to contain negatively correlated objectives especially when the number of objective functions is very large.

To solve this problem, a method based on fuzzy semantics is proposed. Two groups of thresholds are selected in the first instance to define the fuzzy correlation semantic information based on some experiments which are compared with users' perceptions. It is noteworthy the threshold values may be different for different users and thus they can be modified. The first group is used to identify weak correlations and the default threshold values of -0.2 and 0.2 are selected based on experience. In the second group, the default threshold values of -0.8 and 0.8 are selected to identify strong correlations. As shown in Fig. 3, the correlation coefficient between B_2 and D_2 in Case C_2 is -0.18, and, in this case, C_2 is regarded as being acceptable. However, according to the rules described above, Case C_3 in Fig. 3 is regarded as being unacceptable. Next, further adjustments operations are conducted on the weakly correlated objectives to make the objective function cluster more reasonable. The adjustment process is as follows:

- a) Find two unsatisfactory objective points in an objective point cluster and calculate the distances between these two points and the other objective points in the cluster;
- b) Choose the point with the smallest distance value as the one to be adjusted and remove it from the cluster. For example, in Case C_3 in Fig. 3, D_3 will be adjusted;

Fig. 3. Different cases of objective point clusters.

- c) Find the most suitable objective point cluster for the objective point chosen. If there is a suitable cluster, add the objective point into it; otherwise, the objective point will not be adjusted because it is the closest to its original objective point cluster. This point is then labeled as an adjusted point and will be dealt with later.
- d) Convert the objective points in an objective point cluster with multiple objective functions into a single objective during the optimization process. The weight assigned to each objective function can be determined using various methods. In this study, it is calculated with the help of the test population, and in this way, each objective can have a definite effect on the final single objective.

V. EVOLUTION OF A POPULATION

To enable each particle to move towards the true Pareto front and attain a more uniform distribution for solutions in a population after iteration operation, an effective flight guiding strategy is proposed in this research. In the current population, the particles on the current Pareto front are of great interest as they are located in better positions than other particles. The main difference between the proposed GSPSO method and other evolutionary algorithms is that the geometric structure of the Pareto front is exploited to guide the flight of particles in a population. Therefore, it is not necessary for GSPSO to adopt the mechanism of ranking the current Pareto front or the local and global best solutions during each time of iteration. Such a geometric structure is represented using a parametric

(a). The atomic mapping between a mesh triangle and its corresponding parameter triangle
(b). Angles for the discrete harmonic mapping and the mean value coordinates
Fig. 4. Example of a mesh triangle.

equation in the form of an NURBS curve or an NURBS surface from which more information can be obtained to guide the flight of the particles on the current Pareto front.

A. Construction of NURBS Equations for The Pareto Front

The parameterization process aims to transform the current Pareto front from a representation in the objective function space to a representation in the NURBS parameter space. The geometric structure information can then be easily obtained by specifying different parameter values. If the number of objectives is two, an NURBS curve model will be used and a series of values of the parameter u need to be obtained. If the number of objectives is three, an NURBS surface model will be used and a series of values of the parameter (u, v) pair need to be obtained. In this paper, only the process of constructing a parametric equation for three objectives is detailed as the process for two objectives is very similar.

The most important task in this transformation process is to map a given disk-like surface $S \subset \mathbb{R}^3$ into the plane space. With the triangular mesh S_T , the goal is to find a polygonal domain $S^* \subset \mathbb{R}^2$ and a suitable piecewise linear mapping $f: S_T \rightarrow S^*$, as shown in Fig. 4. In this work, the discrete harmonic mapping method [20] is adopted, which involves

two main steps, namely fixing the boundary mapping and finding the piecewise linear mapping. The main advantage of this method is that it is a quadratic minimization problem and reduces the complexity of solving a linear system. For a triangle $T = [v_1 + v_2 + v_3]$ in surface S_T , Equation (8) can be obtained according to Fig. 4(a):

$$2 \int_T |\nabla f|^2 = \cot \theta_1 (f(v_1) - f(v_2))^2 + \cot \theta_2 (f(v_2) - f(v_3))^2 + \cot \theta_3 (f(v_3) - f(v_1))^2 \quad (8)$$

The normal equations for the minimization problem can therefore be expressed as a linear system of equations where $\omega_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$:

$$\sum_{j \in N_i} \omega_{ij} (f(v_j) - f(v_i)) = 0 \quad (9)$$

The angles α_{ij} and β_{ij} are shown in Fig. 4(b). The vertices are assumed to be indexed and N_i is used to denote the set of indexes indicating the neighbors of vertex v_i (i.e. those vertices sharing an edge with v_i).

One linear equation (Equation 9) is constructed for each vertex and thus a linear system of equations can be obtained. It is clear that the associated matrix of the linear system of equations is symmetric and positive definite. Therefore, the linear system is uniquely solvable. It should be noted that the system has to be solved twice: one for u and another one for v for the parameter point $f(v_i)$. Moreover, the case of a sphere-shape Pareto front is not likely since a domination relationship exists between the dominating solutions and the dominated solutions.

After the current Pareto front is mapped into the geometric space, it can be fitted with a NURBS model. The detailed fitting process out the scope of this paper and has been published elsewhere [21].

B. Generation of The Guiding Point Set

The main advantage of GSPSO is achieved by exploiting the geometric structure information e.g. the normal directions and the positions of the particles in the current Pareto front. This information can help the particles make more effective moves during their flights. For instance, the normal directions provide heuristics for the particles about the possible flight directions. A corresponding guiding point is obtained in this study for each particle on the current Pareto front to guide the dominated particles' flights. The process of generating the guiding point for each particle is as follows:

- 1) Obtain the normal vector of each particle based on the geometric structure information. As shown in Fig. 5(a), the normal vector can be obtained by rotating the tangent vector of a particle's NURBS curve by 90° . It should be noted that the partial unit tangent vectors in the u and v directions must be solved if it is a NURBS surface. The product of these two partial unit tangent vectors is then calculated to obtain the normal vector.
- 2) Find out the corresponding guiding point for each particle. The guiding point exists in the normal vector direction of the current particle and the distance between them can be calculated using the following equation :

$$AD = \alpha \sqrt{\sum_{i=1}^N f_i^2} \quad (10)$$

In Equation (10), N represents the number of objectives and its value is either two or three; the radical sign part is the norm of the current particle on the current Pareto front in the objective space in order to consider the magnitudes of the objectives; and α is a coefficient for controlling the position of the guiding point.

- 3) Analyze the domination relationship between the particle and its guiding point. If the current particle is dominated, the normal direction should be adjusted to the opposite direction and the guiding point should be recalculated. Take Fig. 5(a) as an example, normal direction $Normal_1$ cannot be used and has to be adjusted to $Normal_2$.

After all the guiding points are obtained for the particles on the current Pareto front, a guiding point set is obtained as the point set $(A_g, B_g, C_g, D_g, E_g)$, as shown in Fig. 5(b). The computational efficiency of generating the guiding points is actually very high since it is very efficient for a NURBS curve or surface to solve the tangent vector with a given parameter value.

(a) Two normal vectors (b) The guiding point set
Fig. 5. Generation of the guiding point set.

C. Guiding The Flights of Particles

During each time of iteration, GSPSO only deals with the particles that are not located on the Pareto front of the current generation. The main idea is to guide these particles to fly to better regions in which the objective values are more satisfactory using the guiding point set in the objective space. The particles' positions are continuously updated with a variable flying speed while they can be moved to the better region in a few steps. The equations used to update the flight speed are similar to the traditional PSO, as shown in Equations (11) and (12):

$$\begin{aligned} \vec{v}_i(t+1) = & \alpha(\omega \vec{v}_i(t) + C_1(\vec{x}_{p_1} - \vec{x}_i) + C_2(\vec{x}_{p_2} - \vec{x}_i)) \\ & + (1-\alpha)(\vec{x}_g - \vec{x}_i) \end{aligned} \quad (11a)$$

$$\begin{aligned} \vec{v}_i(t+1) = & \alpha(\omega \vec{v}_i(t) + C_1(\vec{x}_{p_1} - \vec{x}_i) + C_2(\vec{x}_{p_2} - \vec{x}_i) \\ & + C_3(\vec{x}_{p_3} - \vec{x}_i)) + (1-\alpha)(\vec{x}_g - \vec{x}_i) \end{aligned} \quad (11b)$$

$$\alpha = 1 - \frac{Iteration}{MaxIteration} \quad (12)$$

Specifically, Equation (11a) is used for NURBS curves while Equation (11b) is for NURBS surfaces. In Equation (11a), ω is the inertia weight of the particle; $\vec{x}_{p_1}$ and $\vec{x}_{p_2}$ are the nearest non-dominated particles; and C_1 and C_2 are

used to control the influences of these two non-dominated particles. As the flight direction is definite, ω is assigned a small value of 0.1. Therefore, no random factors are appended to the two non-dominated particles. Randomness of GSPSO is thus decreased and evolution of individuals is conducted more effectively. $\vec{x}_g$ is a global non-dominated particle randomly selected from the non-dominated archive. The variable *Iteration* is the current iteration number whereas *MaxIteration* is the maximum number of iteration. The coefficient α is used to control the influences of the guiding points and the global particle since it is difficult to find out the precise phase of the current step in a whole evolution process. It can be seen that particle velocity predominantly depends on the guiding points in an early phase. As the evolution process proceeds, the value of coefficient α gradually becomes smaller - this means particle velocity is mainly determined by the global non-dominated particle in a late phase. The case with three objectives also has this characteristic. This flight-guiding method proves to be simple and efficient with the position of each particle updated according to Equation (1).

Based on the above discussion, the local optimization algorithm is proposed and described as follows. Without loss of generality, a case with two objectives is used to explain the algorithm. Any particle's speed is firstly initialized as 0, and the following steps will be executed after the initialization.

- 1) Compute the Euclidian distances between the current dominated particle and the particles on the current Pareto front. Choose two non-dominated particles that have the smallest distances to the current particle and obtain their corresponding guiding points.
- 2) Compute the sum of the distances d_1 between the current particle and the nearest non-dominated particles on the current Pareto front. Take Fig. 6 as an example, the non-dominated particles A and C are the nearest particles of the current particle P, and A_g and C_g are their corresponding guiding points, respectively. The region formed by points A, C, A_g and C_g is then considered to be a better region, which is also the preferable region of P.
- 3) Update the flight speed of P using the non-dominated particles A and C according to Equation (11). In addition to this, update the values of the optimization variables according to Equation (1) and reevaluate the particles concerned.
- 4) Compute the sum of distances d_2 between the particle and the two guiding points.
- 5) If d_1 is less than d_2 , P is not located in the better region. In this case, Step 2 will be executed again. Otherwise, the local optimization process terminates.

The dominated particle can be moved into the better region in a number of steps with the help of the two deterministic non-dominated particles and their corresponding guiding points. In this study, two nearest non-dominated particles are chosen to ensure that the new population is more uniform. In addition, a default maximum iteration number $\mathcal{E}$ is given to avoid unlimited inner loops. By doing several experiments on the test problems, $\mathcal{E}$ is set to be 10 and this value proves to obtain a satisfactory result and achieve good efficiency of computation.

Fig. 6. The local optimization process.

After the local optimization process is executed for all the dominated particles, all the particles in the current population are moved into the better region and the current Pareto front is also updated. As shown in Fig. 7(a), the entire population is composed of the white Pareto front particles and the blue dominated particles while the red points represent the guiding point set of the current Pareto front. Fig. 7(b) shows the particles distribution after one time of iteration is completed when the dominated particles are located in the better region and the Pareto front is updated (the new Pareto front is indicated using the blue curve).

(a). Distribution of the particles in the current population

(b). Distribution of the particles in the next generation

Fig. 7. An example of population evolution.

VI. GENERATION OF A UNIFORM PARETO FRONT FOR THE CURRENT GENERATION

Good uniformity of the Pareto front is important for MOO problems. It indicates that more effective solutions exist and thus engineers have more alternatives for solving a practical engineering design problem. In addition, in some complex MOO problems, designers may need to seek an extremely good value for one aspect even though this will have adverse effects on other aspects. In this case, the objective is often located in an infeasible region.

It is very difficult for traditional MOO algorithms to obtain an ideal uniform Pareto front because the density distribution of a Pareto front is mostly analyzed by calculating the Euclidian distance in the objective space which cannot ensure uniformity of the Pareto front space. Even if the uniform Pareto front can be obtained by inserting some particles into the Pareto front, another challenge is how to obtain the values for the optimization variables of the inserted particles. In this study, the parametric equations constructed for representing the geometric structure are used to solve the first problem.

Meanwhile, a trust region based method is also adapted in this work to solve the second problem.

A. A General Process of Uniformity Operation

Uniformity improvement in the GSPSO method refers to utilizing the parameter information in the geometric space to analyze the density distribution of the Pareto front. And then the Pareto front is made more uniform by adding or removing some particles. If the Pareto front is 2D in the objective space, only the parameter u in the NURBS curve model needs to be considered while the parameters u and v in the NURBS surface model should both be taken into consideration if the number of dimensions in the objective space is three.

Without loss of generality, the distribution of particles on a Pareto front is given in Fig. 8 for the case of two objectives. Obviously, S_1 and S_3 are dense regions whereas regions S_2 and S_4 are sparse regions - so S_2 and S_4 need to be made denser. The general process of inserting a particle is given as follows:

- Calculate the distances for every two neighbor particles based on their u (and v) value(s) and then choose two particles whose distances are the biggest.
- Assign the average u (and v) value(s) of the two chosen particles to that (those) of the inserted particle. Calculate the NURBS curve/surface equation with the specific u (and v) value(s) to obtain the corresponding objective values.
- Calculate the optimization variables using the trust region method to approximate the objectives, which will be discussed in the next subsection.

Fig. 8. Density distribution of a Pareto front.

B. Estimation of Optimization Variable Values Based on Trust Region

After the uniformity operation is completed on the current Pareto front, the next task is to calculate the objective values using the optimization variables. Usually, there are several variables for a given optimization objective. Therefore, it is an approximation problem in which a series of optimization variable values can be obtained, and the objective expressions are evaluated to find out specific values for the optimization variables. In this study, the trust region method proposed by Powell and Yuan [18] is adapted because of its fast speed of convergence. This problem is formalized as follows:

$$\begin{aligned} F(x) &= \mathbf{0} \\ x &\in \Omega = \{x \mid l \leq x \leq u\} \end{aligned} \quad (13)$$

$F: \mathcal{X} \rightarrow \mathbb{R}^m$ is a continuous and differentiable mapping. Equation (13) can be further transformed into the solution of a minimum problem:

$$\begin{aligned} \min f(x) &= \frac{1}{2} \|F(x)\|, \\ \text{s.t. } x &\in \Omega = \{x \mid l \leq x \leq u\} \end{aligned} \quad (14)$$

Powell and Yuan [18] defined several key terms for the trust region method, namely the sub-problem, the criterion for accepting the direction and the step size marked as s_k , the criterion for revising the radius of the trust region marked as Δ_k and the principle of convergence. The direction and the step size are determined simultaneously in the method. These criteria and principles ensure that GSPSO is preferable to other linear search methods. The sub-problem in the trust region method is defined as follows:

$$\begin{aligned} \min q_k(s) &= g_k^T s + \frac{1}{2} s^T B_k s \\ \text{s.t. } \|s\| &\leq \Delta_k \end{aligned} \quad (15)$$

In Equation (15), $s = x - x_k$, $g_k = \nabla f(x_k)$, B_k is the Hesse matrix $\nabla^2 f(x_k)$; Δ_k is the trust region radius; and $\|\cdot\|$ is the norm. Let s_k be the solution of the sub-problem, then the drop of the objective function is given in the following form:

$$Ared_k = f(x_k) - f(x_k + s_k) \quad (16)$$

The drop of the model function is defined as follows:

$$Pred_k = q_k(0) - q_k(s_k) \quad (17)$$

The first drop is the actual change and the latter is the predicted drop. The ratio between them is defined as follows:

$$\rho_k = \frac{Ared_k}{Pred_k} = \frac{f(x_k) - f(x_k + s_k)}{q_k(0) - q_k(s_k)} \quad (18)$$

Equation (18) is in essence used to estimate the degree of consistency between the model function q_k and the objective function f . The trust region radius Δ_k can then be adapted based on the degree of consistency.

VII. COMPUTATIONAL EXPERIMENTS

The proposed GSPSO algorithm is implemented using the Java programming language in this research. On this basis, a number of computational experiments have been conducted to demonstrate the process of using the GSPSO algorithm as well as to evaluate its performance. The experiments mainly consist of four parts: (1) a comparison between GSPSO and four typical algorithms when they are applied to three suites of classic optimization problems; (2) an evaluation of the proposed optimization algorithms in terms of convergence speed; (3) an analysis of the performance and influence of dimension reduction in the GSPSO algorithm; and (4) a demonstration of the application of the proposed algorithm in a practical multi-objective optimization problem, i.e. an electric vehicle.

A. Performance Evaluation

To evaluate the performance of GSPSO, it is compared with four classic multi-objective algorithms, namely NSGAII, MOCell, AbYSS and pMOEAD [27], on the standard MOO problem suites of ZDT (Zitzler-Deb-Thiele) and DTLZ (Deb-Thiele-Laumanns-Zitzler) [28]. Amongst these algorithms, pMOEAD is actually a newer version of MOEAD and is given such a name in this paper to make a distinction between the two. In addition, a more recent suite with the unconstrained functions from the IEEE CEC 2009 competition [29] is also used in the experiments. All the experiment results are analyzed and compared using the following four performance indicators, namely the HV (Hyper Volume), the additive unary EPSILON $I_{\epsilon+}^1$ [30], SPREAD Δ and the IGD (Inverted Generational Distance). These indicators are mainly used to assess convergence speed and the quality of the Pareto fronts obtained. The larger HV is, the better the optimal Pareto front is. For the other indicators, the smaller they are, the closer the corresponding result is to the true Pareto front.

The common parameters for all the algorithms are set to be the same. For example, the population size is set to 100 and the archive size is set to 100. Additionally, the maximum number of iteration for evaluating the objective functions for these algorithms is set to 25000. For NSGAII, the internal population size is set to 100; the crossover probability is $p_c = 0.9$ and the mutation probability is $p_m = 1.0/L$ where L is the number of optimization variables.

As random exploration is a typical feature for evolutionary algorithms, each algorithm is run 50 times for each test. The experiments results are analyzed by using statistical values such as the mean value, standard value, median value and interquartile range value. The statistical values for HV are shown from Fig. 9 to Fig. 12 using histograms. It's intuitive from these figures that GSPSO can achieve good performances for almost test problems (the higher for the histograms, the better values for HV). The statistical values for other performance indicators are shown in Tables I through to VI. Specifically, Table I shows the mean and standard deviation of Δ ; Table II shows the median and IQR of Δ ; Table III shows the mean and standard derivation of $I_{\epsilon+}^1$; Table IV shows the median and IQR of $I_{\epsilon+}^1$; Table V shows the mean and standard derivation of IGD; and Table VI shows the median and IQR of IGD. The best solution in the tables is highlighted in dark gray while the second best solution is highlighted in light gray. As for HV indicator, it will be presented and discussed in Boxplots and paired samples t-test in Section B.2, thus no comparison is presented here.

From these tables, a number of observations can be made. Firstly, GSPSO can achieve the best performance in most of the test problems in terms of the four indicators. This means GSPSO can best approximate to the true Pareto front while achieving an excellent uniformity. Nevertheless, it cannot obtain a good optimal Pareto front for ZDT6, DTLZ2, DTLZ7, UF5, UF6 and UF9. Therefore, it is very interesting and useful to investigate the characteristics of the above tests and make improvements to GSPSO to better solve these

problems. Secondly, NSGAII and MOCELL perform the second best in a few tests including ZDT3, DTLZ1, DTLZ2, DTLZ4, DTLZ5, DTLZ7, UF1, UF2, UF4, UF5, UF7 and UF8. They can attain an excellent approximation for most of problems in the DTLZ suite and the UF suite, as evidenced in the $I_{\varepsilon+}^1$ and IGD indicators in Tables V through to VIII. Thirdly, no obvious overall differences have been found in the performances of NSGAII and MOCell. However, they do get slightly different statistical results in terms of some particular indicators. For example, NSGAII achieves good results in $I_{\varepsilon+}^1$ and IGD, confirming its effectiveness in terms of approximation for some test problems. MOCell obtains good values of Δ for some problems, which indicates a good uniformity is attained.

Table I and Table II show that GSPSO achieves the best performance in terms of the Δ indicator in most of the test problems with the help of the uniformity operation. GSPSO also gives satisfactory results on the other four optimization problems. Specifically, MOCell performs well in problems ZDT6, DTLZ2, DTLZ5, DTLZ7, UF8 and UF9. It thus can be concluded that MOCell achieves the second best performance in the experiments. PMOEAD only performs well in problems DTLZ1 and UF6 but has unsatisfactory performance for the other problems. Even though NSGAII and AbYSS achieve the second best results in some test problems, they have worse performance compared with other algorithms in terms of uniformity. Therefore, these three algorithms should be greatly improved in this aspect.

Tables III and IV show the statistical results of $I_{\varepsilon+}^1$ and Tables V and VI show results for the IGD indicator. Both of the indicators are used to evaluate how well the results obtained approximate the optimal Pareto front. It can be seen that GSPSO achieves the best performance in approximation terms for most of the problems. This is mainly ascribed to it taking advantage of the geometric structure information of the Pareto front. GSPSO does not have satisfactory performance for problems DTLZ6, UF5, UF6, UF8 and UF9. Therefore, it should be improved to make it more suitable for these problems in particular the UF test suite. NSGAII achieves the second best performance in most tests especially for the DTLZ and UF suites.

Fig. 9. Statistics results of HV mean values.

Fig. 10. Statistics results of HV standard deviation values.

Fig. 11. Statistics results of HV median values.

Fig. 12. Statistics results of HV IQR values.

TABLE I
SPREAD: MEAN AND STANDARD DEVIATION

	NSGAII	pMOEAD	MOCeII	AbYSS	GSPSO
ZDT1	3.72e-01 _{3.2e-02}	8.35e-01 _{4.1e-02}	8.35e-02 _{1.6e-02}	5.63e-01 _{1.8e-01}	5.84e-02 _{1.0e-02}
ZDT2	3.78e-01 _{3.4e-02}	1.01e+00 _{5.3e-02}	8.00e-02 _{1.3e-02}	5.28e-01 _{2.1e-01}	6.00e-02 _{1.3e-02}
ZDT3	7.47e-01 _{1.2e-02}	9.92e-01 _{4.5e-02}	7.10e-01 _{2.0e-02}	8.97e-01 _{6.1e-02}	7.01e-01 _{3.0e-04}
ZDT4	3.95e-01 _{3.7e-02}	1.17e+00 _{7.4e-02}	1.30e-01 _{4.6e-02}	9.35e-01 _{1.5e-01}	6.41e-02 _{1.3e-02}
ZDT6	3.63e-01 _{2.9e-02}	8.47e-01 _{4.0e-01}	1.05e-01 _{1.1e-02}	4.52e-01 _{1.5e-01}	1.37e-01 _{2.4e-01}
DTLZ1	9.57e-01 _{2.3e-01}	9.56e-01 _{1.8e-01}	1.09e+00 _{2.1e-01}	9.81e-01 _{1.2e-01}	6.75e-01 _{3.7e-02}
DTLZ2	7.03e-01 _{4.5e-02}	8.26e-01 _{2.8e-02}	6.92e-01 _{4.2e-02}	7.43e-01 _{4.9e-02}	6.22e-01 _{4.2e-02}
DTLZ3	1.07e+00 _{1.1e-01}	1.08e+00 _{1.9e-01}	1.16e+00 _{1.3e-01}	1.03e+00 _{9.6e-02}	6.37e-01 _{3.5e-02}
DTLZ4	6.90e-01 _{5.6e-02}	1.04e+00 _{1.8e-01}	7.45e-01 _{1.6e-01}	7.11e-01 _{4.8e-02}	6.28e-01 _{4.2e-02}
DTLZ5	4.54e-01 _{4.9e-02}	7.35e-01 _{2.5e-02}	1.47e-01 _{3.5e-02}	3.31e-01 _{1.2e-01}	1.30e-01 _{3.8e-02}
DTLZ6	8.13e-01 _{5.5e-02}	7.31e-01 _{9.9e-03}	7.46e-01 _{4.2e-02}	7.59e-01 _{4.0e-02}	1.05e-01 _{1.1e-02}
DTLZ7	7.42e-01 _{4.9e-02}	1.02e+00 _{9.0e-02}	7.21e-01 _{5.0e-02}	7.28e-01 _{6.0e-02}	7.21e-01 _{4.7e-02}
UF1	1.47e+00 _{1.3e-01}	1.07e+00 _{9.0e-02}	1.42e+00 _{2.1e-01}	1.43e+00 _{1.9e-01}	8.63e-01 _{1.1e-01}
UF2	6.35e-01 _{6.6e-02}	7.44e-01 _{8.3e-02}	5.46e-01 _{7.5e-02}	8.94e-01 _{1.0e-01}	3.62e-01 _{6.9e-02}
UF3	1.03e+00 _{6.2e-02}	1.02e+00 _{1.0e-01}	1.09e+00 _{9.1e-02}	1.02e+00 _{6.0e-02}	3.88e-01 _{1.2e-01}
UF4	5.37e-01 _{5.7e-02}	5.02e-01 _{9.3e-02}	5.52e-01 _{1.0e-01}	6.76e-01 _{7.0e-02}	2.84e-01 _{4.6e-02}
UF5	1.20e+00 _{9.4e-02}	1.07e+00 _{6.2e-02}	1.13e+00 _{1.4e-01}	1.10e+00 _{1.3e-01}	9.93e-01 _{1.1e-01}
UF6	1.16e+00 _{1.3e-01}	1.06e+00 _{7.8e-02}	1.19e+00 _{1.7e-01}	1.20e+00 _{1.7e-01}	1.08e+00 _{1.1e-01}
UF7	1.05e+00 _{1.7e-01}	9.67e-01 _{1.2e-01}	1.08e+00 _{1.4e-01}	1.17e+00 _{1.2e-01}	6.07e-01 _{9.8e-02}
UF8	7.81e-01 _{8.7e-02}	8.17e-01 _{1.3e-01}	7.41e-01 _{8.6e-02}	8.03e-01 _{1.1e-01}	7.63e-01 _{7.3e-02}
UF9	8.32e-01 _{7.2e-02}	9.42e-01 _{9.5e-02}	7.63e-01 _{5.9e-02}	7.77e-01 _{7.1e-02}	8.38e-01 _{4.9e-02}
UF10	7.85e-01 _{7.9e-02}	9.41e-01 _{4.9e-02}	8.32e-01 _{1.1e-01}	8.17e-01 _{1.3e-01}	7.49e-01 _{8.6e-02}

TABLE II
SPREAD: MEDIAN AND IQR

	NSGAII	pMOEAD	MOCeII	AbYSS	GSPSO
ZDT1	3.72e-01 _{3.8e-02}	8.40e-01 _{5.5e-02}	8.17e-02 _{2.0e-02}	5.71e-01 _{2.9e-01}	5.72e-02 _{1.4e-02}
ZDT2	3.72e-01 _{3.1e-02}	1.01e+00 _{2.6e-02}	8.00e-02 _{2.0e-02}	4.84e-01 _{2.9e-01}	6.03e-02 _{1.9e-02}
ZDT3	7.44e-01 _{1.8e-02}	9.92e-01 _{5.5e-02}	7.07e-01 _{7.0e-03}	8.88e-01 _{8.5e-02}	7.01e-01 _{3.4e-04}
ZDT4	3.99e-01 _{4.5e-02}	1.18e+00 _{9.2e-02}	1.24e-01 _{2.4e-02}	9.62e-01 _{1.4e-01}	6.46e-02 _{1.7e-02}
ZDT6	3.57e-01 _{4.2e-02}	9.46e-01 _{6.9e-01}	1.04e-01 _{1.3e-02}	4.31e-01 _{2.0e-01}	5.03e-02 _{4.3e-02}
DTLZ1	8.75e-01 _{1.3e-01}	8.84e-01 _{1.7e-01}	1.05e+00 _{2.3e-01}	9.45e-01 _{1.7e-01}	6.68e-01 _{5.2e-02}
DTLZ2	7.04e-01 _{4.2e-02}	8.26e-01 _{3.7e-02}	6.90e-01 _{5.3e-02}	7.43e-01 _{6.7e-02}	6.18e-01 _{6.5e-02}
DTLZ3	1.06e+00 _{1.2e-01}	1.03e+00 _{2.2e-01}	1.14e+00 _{1.9e-01}	1.03e+00 _{1.4e-01}	6.36e-01 _{4.2e-02}
DTLZ4	6.85e-01 _{8.6e-02}	9.97e-01 _{3.6e-01}	6.65e-01 _{3.6e-01}	7.11e-01 _{7.1e-02}	6.30e-01 _{5.7e-02}
DTLZ5	4.56e-01 _{7.4e-02}	7.37e-01 _{4.7e-02}	1.37e-01 _{3.4e-02}	3.06e-01 _{1.9e-01}	1.22e-01 _{5.8e-02}
DTLZ6	8.03e-01 _{8.5e-02}	7.32e-01 _{1.1e-02}	7.51e-01 _{5.8e-02}	7.60e-01 _{4.3e-02}	1.06e-01 _{1.5e-02}
DTLZ7	7.47e-01 _{6.2e-02}	1.03e+00 _{1.2e-01}	7.16e-01 _{7.1e-02}	7.34e-01 _{7.7e-02}	7.19e-01 _{6.3e-02}
UF1	1.48e+00 _{1.4e-01}	1.08e+00 _{1.5e-01}	1.49e+00 _{2.7e-01}	1.50e+00 _{2.0e-01}	8.56e-01 _{1.6e-01}
UF2	6.32e-01 _{8.4e-02}	7.12e-01 _{1.1e-01}	5.25e-01 _{8.7e-02}	8.83e-01 _{1.2e-01}	3.68e-01 _{8.6e-02}
UF3	1.02e+00 _{1.0e-01}	1.01e+00 _{1.4e-01}	1.07e+00 _{1.5e-01}	1.01e+00 _{5.6e-02}	3.70e-01 _{1.5e-01}
UF4	5.37e-01 _{8.1e-02}	4.79e-01 _{1.3e-01}	5.38e-01 _{1.3e-01}	6.85e-01 _{1.0e-01}	2.83e-01 _{5.6e-02}
UF5	1.22e+00 _{1.6e-01}	1.07e+00 _{1.0e-01}	1.07e+00 _{1.9e-01}	1.10e+00 _{1.7e-01}	9.89e-01 _{1.1e-01}
UF6	1.13e+00 _{1.4e-01}	1.03e+00 _{1.0e-01}	1.13e+00 _{2.7e-01}	1.19e+00 _{2.7e-01}	1.08e+00 _{1.3e-01}
UF7	1.03e+00 _{2.8e-01}	9.66e-01 _{1.7e-01}	1.01e+00 _{2.0e-01}	1.19e+00 _{1.1e-01}	6.09e-01 _{1.0e-01}
UF8	7.72e-01 _{1.3e-01}	7.89e-01 _{9.2e-02}	7.19e-01 _{1.1e-01}	8.03e-01 _{1.5e-01}	7.53e-01 _{1.1e-01}
UF9	8.34e-01 _{1.0e-01}	9.40e-01 _{9.5e-02}	7.64e-01 _{8.7e-02}	7.60e-01 _{8.9e-02}	8.49e-01 _{7.3e-02}
UF10	7.83e-01 _{1.0e-01}	9.46e-01 _{5.5e-02}	8.32e-01 _{1.7e-01}	7.97e-01 _{1.4e-01}	7.57e-01 _{8.3e-02}

TABLE III
EPSILON: MEAN AND STANDARD DEVIATION

	NSGAII	pMOEAD	MOCeII	AbYSS	GSPSO
ZDT1	1.38e-02 _{1.9e-03}	4.62e-01 _{1.3e-01}	6.65e-03 _{3.9e-04}	9.02e-02 _{5.3e-02}	5.15e-03 _{1.1e-04}
ZDT2	1.35e-02 _{1.8e-03}	1.28e+00 _{2.0e-01}	5.87e-03 _{2.9e-04}	1.31e-01 _{1.7e-01}	5.11e-03 _{1.1e-04}
ZDT3	1.45e-02 _{4.3e-02}	9.08e-01 _{2.1e-01}	6.92e-02 _{1.4e-01}	2.02e-01 _{1.1e-01}	4.79e-03 _{1.3e-04}
ZDT4	1.57e-02 _{3.2e-03}	4.84e+00 _{1.9e+00}	9.94e-03 _{4.2e-03}	6.26e-01 _{4.0e-01}	5.17e-03 _{1.3e-04}
ZDT6	1.45e-02 _{1.8e-03}	2.77e-01 _{2.5e-01}	7.89e-03 _{6.3e-04}	5.38e-02 _{1.8e-02}	4.43e-03 _{4.6e-04}
DTLZ1	1.12e-01 _{8.9e-02}	5.28e-01 _{9.7e-01}	4.52e-01 _{4.0e-01}	1.17e+00 _{8.2e-01}	5.69e-02 _{5.5e-03}
DTLZ2	1.32e-01 _{1.7e-02}	1.50e-01 _{6.6e-03}	1.33e-01 _{1.7e-02}	1.46e-01 _{1.6e-02}	1.32e-01 _{1.8e-02}
DTLZ3	5.28e+00 _{2.4e+00}	1.71e+01 _{1.9e+01}	1.05e+01 _{5.4e+00}	2.73e+01 _{1.2e+01}	1.32e-01 _{1.7e-02}
DTLZ4	1.11e-01 _{1.7e-02}	1.71e-01 _{5.7e-02}	3.50e-01 _{3.9e-01}	1.22e-01 _{1.8e-02}	1.09e-01 _{1.5e-02}
DTLZ5	1.14e-02 _{2.4e-03}	2.31e-02 _{9.9e-04}	5.62e-03 _{3.6e-03}	1.48e-02 _{8.0e-03}	4.19e-03 _{1.9e-04}
DTLZ6	8.80e-01 _{7.2e-02}	2.03e-02 _{3.4e-04}	1.71e+00 _{1.3e-01}	2.69e+00 _{2.9e-01}	4.12e-03 _{2.4e-04}
DTLZ7	1.36e-01 _{5.2e-02}	1.52e+00 _{7.8e-01}	6.79e-01 _{1.5e-01}	8.99e-01 _{3.3e-01}	1.88e-01 _{5.7e-02}
UF1	2.23e-01 _{5.8e-02}	2.55e-01 _{6.4e-02}	3.25e-01 _{1.2e-01}	2.67e-01 _{7.2e-02}	1.14e-01 _{2.3e-02}
UF2	1.20e-01 _{1.8e-02}	2.14e-01 _{5.1e-02}	2.04e-01 _{7.7e-02}	1.79e-01 _{2.9e-02}	6.82e-02 _{7.1e-03}
UF3	3.58e-01 _{5.2e-02}	4.34e-01 _{1.0e-01}	5.23e-01 _{9.5e-02}	5.38e-01 _{7.5e-02}	1.53e-01 _{3.8e-02}
UF4	7.23e-02 _{8.2e-03}	1.04e-01 _{7.0e-03}	8.55e-02 _{8.4e-03}	1.03e-01 _{5.8e-03}	6.29e-02 _{7.3e-03}
UF5	7.66e-01 _{2.2e-01}	1.62e+00 _{2.3e-01}	8.20e-01 _{2.1e-01}	1.19e+00 _{2.9e-01}	1.40e+00 _{4.1e-01}
UF6	4.48e-01 _{1.4e-01}	8.77e-01 _{1.9e-01}	6.61e-01 _{2.8e-01}	6.13e-01 _{2.3e-01}	5.18e-01 _{1.3e-01}
UF7	3.63e-01 _{2.4e-01}	4.39e-01 _{2.6e-01}	6.69e-01 _{2.4e-01}	5.19e-01 _{2.1e-01}	1.04e-01 _{7.7e-02}
UF8	6.57e-01 _{1.2e-01}	7.40e-01 _{2.0e-02}	7.19e-01 _{1.1e-01}	8.46e-01 _{1.1e-01}	6.88e-01 _{1.0e-01}
UF9	5.32e-01 _{8.8e-02}	5.52e-01 _{8.6e-02}	5.42e-01 _{5.6e-02}	6.78e-01 _{1.2e-01}	6.95e-01 _{8.2e-02}
UF10	1.02e+00 _{1.5e-01}	1.77e+00 _{2.5e-01}	8.86e-01 _{8.7e-02}	1.87e+00 _{7.0e-01}	7.88e-01 _{8.1e-02}

TABLE IV
EPSILON: MEDIAN AND IQR

	NSGAII	pMOEAD	MOCcell	AbYSS	GSPSO
ZDT1	1.39e-02 _{2.5e-03}	4.65e-01 _{2.4e-01}	6.60e-03 _{4.4e-04}	7.46e-02 _{6.6e-02}	5.14e-03 _{1.5e-04}
ZDT2	1.34e-02 _{2.4e-03}	1.29e+00 _{2.1e-01}	5.82e-03 _{3.3e-04}	5.73e-02 _{6.6e-02}	5.11e-03 _{1.7e-04}
ZDT3	8.04e-03 _{1.5e-03}	9.17e-01 _{2.9e-01}	6.36e-03 _{1.7e-03}	1.71e-01 _{4.8e-02}	4.80e-03 _{1.6e-04}
ZDT4	1.53e-02 _{3.2e-03}	4.70e+00 _{2.8e+00}	9.19e-03 _{2.8e-03}	5.62e-01 _{5.3e-01}	5.16e-03 _{1.6e-04}
ZDT6	1.41e-02 _{2.3e-03}	2.19e-01 _{4.8e-01}	7.87e-03 _{6.2e-04}	4.96e-02 _{2.6e-02}	4.32e-03 _{1.8e-04}
DTLZ1	7.64e-02 _{3.3e-02}	2.42e-01 _{5.0e-01}	3.41e-01 _{4.3e-01}	9.10e-01 _{8.5e-01}	5.63e-02 _{8.1e-03}
DTLZ2	1.31e-01 _{2.2e-02}	1.50e-01 _{5.3e-03}	1.33e-01 _{2.7e-02}	1.45e-01 _{2.0e-02}	1.29e-01 _{2.7e-02}
DTLZ3	5.05e+00 _{2.8e+00}	9.82e+00 _{2.7e+01}	1.01e+01 _{6.6e+00}	2.63e+01 _{1.8e+01}	1.29e-01 _{2.3e-02}
DTLZ4	1.07e-01 _{2.5e-02}	1.42e-01 _{1.2e-01}	1.13e-01 _{8.8e-01}	1.22e-01 _{3.2e-02}	1.06e-01 _{2.1e-02}
DTLZ5	1.09e-02 _{2.9e-03}	2.32e-02 _{1.0e-03}	4.61e-03 _{1.0e-03}	1.28e-02 _{8.6e-03}	4.14e-03 _{1.5e-04}
DTLZ6	8.83e-01 _{8.4e-02}	2.02e-02 _{3.4e-04}	1.72e+00 _{1.2e-01}	2.70e+00 _{3.7e-01}	4.08e-03 _{1.4e-04}
DTLZ7	1.25e-01 _{3.5e-02}	1.46e+00 _{1.3e+00}	1.75e+00 _{1.2e+00}	8.45e-01 _{4.9e-01}	1.71e-01 _{6.9e-02}
UF1	1.97e-01 _{1.0e-01}	2.46e-01 _{7.7e-02}	3.09e-01 _{1.5e-01}	2.86e-01 _{1.2e-01}	1.16e-01 _{2.7e-02}
UF2	1.19e-01 _{2.1e-02}	2.07e-01 _{8.4e-02}	1.87e-01 _{1.5e-01}	1.74e-01 _{4.2e-02}	6.76e-02 _{1.1e-02}
UF3	3.48e-01 _{7.4e-02}	4.04e-01 _{1.7e-01}	5.19e-01 _{1.5e-01}	5.30e-01 _{9.6e-02}	1.59e-01 _{6.1e-02}
UF4	7.05e-02 _{1.1e-02}	1.04e-01 _{9.1e-03}	8.40e-02 _{1.3e-02}	1.02e-01 _{8.4e-03}	6.19e-02 _{9.6e-03}
UF5	7.40e-01 _{3.6e-01}	1.62e+00 _{2.9e-01}	8.15e-01 _{3.5e-01}	1.20e+00 _{4.6e-01}	1.35e+00 _{5.5e-01}
UF6	4.45e-01 _{1.8e-01}	8.82e-01 _{2.7e-01}	6.26e-01 _{4.7e-01}	5.72e-01 _{3.7e-01}	4.71e-01 _{1.5e-01}
UF7	1.94e-01 _{4.8e-01}	2.59e-01 _{5.0e-01}	7.14e-01 _{4.5e-01}	5.92e-01 _{4.3e-01}	9.90e-02 _{3.2e-02}
UF8	7.13e-01 _{9.3e-02}	7.42e-01 _{1.6e-02}	7.18e-01 _{1.1e-02}	8.53e-01 _{1.8e-01}	7.16e-01 _{2.0e-02}
UF9	5.45e-01 _{1.1e-01}	5.14e-01 _{1.1e-01}	5.47e-01 _{8.3e-02}	6.78e-01 _{1.5e-01}	7.12e-01 _{7.5e-02}
UF10	1.00e+00 _{1.4e-01}	1.77e+00 _{3.0e-01}	9.08e-01 _{1.2e-01}	1.82e+00 _{1.1e+00}	7.73e-01 _{1.5e-01}

TABLE V
IGD: MEAN AND STANDARD DEVIATION

	NSGAII	pMOEAD	MOCcell	AbYSS	GSPSO
ZDT1	1.89e-04 _{8.9e-06}	1.18e-02 _{3.3e-03}	1.40e-04 _{1.7e-06}	1.52e-03 _{1.0e-03}	1.34e-04 _{6.1e-07}
ZDT2	1.94e-04 _{9.7e-06}	2.67e-02 _{6.0e-03}	1.41e-04 _{2.3e-06}	1.80e-03 _{2.3e-03}	1.39e-04 _{1.5e-06}
ZDT3	3.51e-04 _{6.8e-04}	2.36e-02 _{4.1e-03}	1.25e-03 _{2.4e-03}	4.26e-03 _{1.9e-03}	1.92e-04 _{1.6e-06}
ZDT4	2.40e-04 _{5.4e-05}	1.43e-01 _{6.0e-02}	1.88e-04 _{4.5e-05}	1.56e-02 _{1.1e-02}	1.34e-04 _{6.8e-07}
ZDT6	3.52e-04 _{3.9e-05}	4.49e-03 _{5.6e-03}	2.17e-04 _{2.0e-05}	1.44e-03 _{4.9e-04}	1.16e-04 _{5.1e-06}
DTLZ1	1.77e-03 _{2.4e-03}	1.21e-02 _{4.4e-02}	9.07e-03 _{9.5e-03}	2.75e-02 _{2.0e-02}	6.32e-04 _{2.4e-05}
DTLZ2	7.69e-04 _{3.9e-05}	8.31e-04 _{1.3e-05}	7.65e-04 _{3.8e-05}	8.27e-04 _{4.7e-05}	7.64e-04 _{3.2e-05}
DTLZ3	9.25e-02 _{4.4e-02}	3.57e-01 _{4.1e-01}	1.67e-01 _{8.9e-02}	4.87e-01 _{2.3e-01}	1.22e-03 _{5.2e-05}
DTLZ4	1.22e-03 _{1.1e-04}	1.29e-03 _{2.9e-04}	3.49e-03 _{3.7e-03}	1.27e-03 _{1.2e-04}	1.28e-03 _{1.0e-04}
DTLZ5	2.05e-05 _{1.4e-06}	3.46e-05 _{1.3e-06}	1.42e-05 _{3.8e-07}	2.39e-05 _{1.3e-05}	1.40e-05 _{2.9e-07}
DTLZ6	7.30e-03 _{6.3e-04}	8.75e-05 _{9.7e-07}	1.49e-02 _{1.3e-03}	2.42e-02 _{2.5e-03}	3.41e-03 _{5.7e-07}
DTLZ7	2.23e-03 _{1.5e-04}	1.78e-02 _{5.2e-03}	8.07e-03 _{7.6e-03}	8.05e-03 _{3.0e-03}	2.81e-03 _{3.2e-04}
UF1	4.64e-03 _{1.1e-03}	4.89e-03 _{1.2e-03}	7.74e-03 _{3.1e-03}	5.77e-03 _{1.5e-03}	2.14e-03 _{3.9e-04}
UF2	1.83e-03 _{3.8e-04}	3.51e-03 _{1.1e-03}	4.01e-03 _{2.2e-03}	3.27e-03 _{4.8e-04}	1.05e-03 _{1.2e-04}
UF3	7.70e-03 _{1.2e-03}	9.40e-03 _{7.9e-04}	1.12e-02 _{1.7e-03}	1.35e-02 _{1.5e-03}	4.39e-03 _{1.5e-03}
UF4	1.94e-03 _{8.2e-05}	3.12e-03 _{3.2e-04}	2.19e-03 _{1.2e-04}	3.01e-03 _{1.2e-04}	1.67e-03 _{1.1e-04}
UF5	1.27e-01 _{4.3e-02}	3.38e-01 _{6.6e-02}	1.24e-01 _{3.6e-02}	2.31e-01 _{6.9e-02}	2.84e-01 _{1.3e-01}
UF6	1.01e-02 _{4.1e-03}	2.09e-02 _{3.4e-03}	1.68e-02 _{7.0e-03}	1.56e-02 _{6.5e-03}	1.46e-02 _{3.3e-03}
UF7	6.88e-03 _{5.8e-03}	8.21e-03 _{5.8e-03}	1.51e-02 _{6.7e-03}	1.05e-02 _{5.1e-03}	1.33e-03 _{1.5e-03}
UF8	2.84e-03 _{3.2e-04}	3.23e-03 _{8.8e-04}	3.11e-03 _{8.2e-04}	4.18e-03 _{1.3e-03}	3.04e-03 _{4.1e-04}
UF9	3.75e-03 _{8.4e-04}	3.31e-03 _{5.1e-04}	3.40e-03 _{5.0e-04}	5.12e-03 _{1.5e-03}	4.67e-03 _{4.9e-04}
UF10	6.41e-03 _{2.8e-03}	1.98e-02 _{3.0e-03}	5.30e-03 _{1.1e-03}	2.03e-02 _{1.1e-02}	4.19e-03 _{5.6e-04}

TABLE VI
IGD: MEDIAN AND IQR

	NSGAII	pMOEAD	MOCcell	AbYSS	GSPSO
ZDT1	1.89e-04 _{1.2e-05}	1.16e-02 _{5.2e-03}	1.40e-04 _{2.6e-06}	1.13e-03 _{1.1e-03}	1.34e-04 _{7.9e-07}
ZDT2	1.92e-04 _{1.3e-05}	2.81e-02 _{5.9e-03}	1.40e-04 _{3.0e-06}	7.81e-04 _{1.3e-03}	1.39e-04 _{2.0e-06}
ZDT3	2.51e-04 _{1.5e-05}	2.40e-02 _{6.6e-03}	2.04e-04 _{7.1e-06}	3.83e-03 _{2.7e-03}	1.92e-04 _{2.2e-06}
ZDT4	2.22e-04 _{5.7e-05}	1.39e-01 _{8.6e-02}	1.76e-04 _{6.6e-05}	1.36e-02 _{1.3e-02}	1.34e-04 _{8.4e-07}
ZDT6	3.51e-04 _{4.7e-05}	2.70e-03 _{5.8e-03}	2.16e-04 _{2.5e-05}	1.33e-03 _{6.8e-04}	1.16e-04 _{7.1e-06}
DTLZ1	7.97e-04 _{4.0e-04}	4.79e-03 _{1.2e-02}	6.50e-03 _{1.1e-02}	2.14e-02 _{2.4e-02}	6.32e-04 _{3.2e-05}
DTLZ2	7.67e-04 _{4.9e-05}	8.29e-04 _{1.9e-05}	7.58e-04 _{3.9e-05}	8.18e-04 _{6.0e-05}	7.65e-04 _{4.7e-05}
DTLZ3	8.88e-02 _{4.8e-02}	2.01e-01 _{5.4e-01}	1.56e-01 _{1.1e-01}	4.62e-01 _{2.7e-01}	1.21e-03 _{9.2e-05}
DTLZ4	1.23e-03 _{1.5e-04}	1.16e-03 _{5.9e-04}	1.24e-03 _{8.3e-03}	1.25e-03 _{1.5e-04}	1.28e-03 _{1.3e-04}
DTLZ5	2.01e-05 _{2.1e-06}	3.44e-05 _{1.5e-06}	1.42e-05 _{4.0e-07}	1.89e-05 _{7.7e-06}	1.39e-05 _{3.7e-07}
DTLZ6	7.23e-03 _{8.9e-04}	8.74e-05 _{1.0e-06}	1.49e-02 _{1.4e-03}	2.44e-02 _{3.7e-03}	3.40e-05 _{9.9e-07}
DTLZ7	2.22e-03 _{1.6e-04}	1.78e-02 _{6.6e-03}	2.52e-03 _{1.3e-02}	7.40e-03 _{4.0e-03}	2.78e-03 _{3.6e-04}
UF1	4.36e-03 _{2.1e-03}	4.69e-03 _{1.8e-03}	7.03e-03 _{4.6e-03}	6.23e-03 _{2.5e-03}	2.08e-03 _{5.7e-04}
UF2	1.74e-03 _{4.5e-04}	3.13e-03 _{1.5e-03}	3.21e-03 _{3.8e-03}	3.20e-03 _{6.5e-04}	1.03e-03 _{1.4e-04}
UF3	7.34e-03 _{1.8e-03}	9.36e-03 _{1.1e-03}	1.13e-02 _{2.0e-03}	1.39e-02 _{2.3e-03}	4.15e-03 _{2.7e-03}
UF4	1.93e-03 _{1.3e-04}	3.17e-03 _{4.0e-04}	2.16e-03 _{1.7e-04}	3.00e-03 _{1.5e-04}	1.66e-03 _{1.5e-04}
UF5	1.21e-01 _{6.0e-02}	3.34e-01 _{6.7e-02}	1.16e-01 _{5.4e-02}	2.25e-01 _{9.7e-02}	2.31e-01 _{1.7e-01}
UF6	9.15e-03 _{3.6e-03}	2.00e-02 _{5.8e-03}	1.58e-02 _{1.1e-02}	1.38e-02 _{1.1e-02}	1.46e-02 _{4.1e-03}
UF7	3.03e-03 _{1.1e-02}	5.09e-03 _{8.7e-03}	1.56e-02 _{8.5e-03}	1.16e-02 _{1.0e-02}	1.17e-03 _{3.3e-04}
UF8	2.87e-03 _{2.7e-04}	3.21e-03 _{3.4e-04}	2.93e-03 _{1.8e-04}	3.76e-03 _{1.3e-03}	3.08e-03 _{2.7e-04}
UF9	3.52e-03 _{1.3e-03}	3.14e-03 _{8.0e-04}	3.29e-03 _{7.1e-04}	4.80e-03 _{2.0e-03}	4.80e-03 _{4.6e-04}
UF10	6.15e-03 _{4.1e-03}	1.99e-02 _{4.2e-03}	5.19e-03 _{1.3e-03}	1.84e-02 _{2.0e-02}	4.07e-03 _{6.6e-04}

The above results provide evidences for the effectiveness of the proposed GSPSO algorithm as a whole. After this, the specific contribution from each component (e.g. dimension reduction) to the overall performance needs to be discussed using the single factor comparison principle. Specifically, three experiments have been conducted.

Experiment 1: The contribution of the evolution strategy exploiting the geometric structure of the Pareto front.

To illustrate the contribution of the geometric structure, the proposed GSPSO is compared with a variant of GSPSO called VGSPSO in which a traditional PSO evolution strategy is used rather than the one using NURBS parametric equations. In addition, the HV indicator is chosen as the main factor for making comparison since the convergence speed is a primary issue of any evolution strategy. The key configuration parameters for running this computational experiment are given as follows: *populationSize* = 100, *maximumEvaluations* = 25000, *numberOfRunning* = 50. The statistical results of HV are shown in Table VII(a) and (b) from which it can be concluded that for most of the test problems GSPSO achieves better convergence performances than VGSPSO. This finding proves that the evolution strategy with NURBS makes an important contribution to the overall performance of the algorithm in terms of convergence speed.

Experiment 2: The contribution of trust region estimation to the overall performance.

To analyze the contribution of the trust region estimation method, GSPSO is compared to a new version of VGSPSO. In this new version, the uniform strategy in classic NSGAI is adopted while other operations are the same as GSPSO. The SPREAD indicator is chosen this time to find the difference between the final optimal fronts obtained in the two algorithms in terms of uniformity. This experiment uses the same configuration parameters as Experiment 1. Table VIII(a) and (b) show the statistical results of SPREAD from which it can be found that GSPSO achieves better distribution. This result confirms the effectiveness of the uniformity strategy with trust region estimation as well as the excellent contribution it makes to the overall performance.

Experiment 3: The contribution of the dimension reduction strategy on the overall performance.

It should be noted that the dimension reduction strategy must only be applied when the number of optimization objectives is over three. In this experiment, GSPSO is again compared with another new version of VGSPSO. In this version, a random method for clustering objectives is adopted while other operations remain unchanged. In this experiment, the focus is on the HV and SPREAD indicators, and the same configuration parameters are used again. Table IX and Table X show the statistical results of HV and SPREAD. It can be seen that GSPSO achieves better performance than VGSPSO. Thus, the dimension reduction strategy employed in GSPSO also makes a very positive contribution to the final overall performances.

TABLE VII

(a) HV: Mean and standard deviation

	VGSPSO	GSPSO
ZDT1	6.60e-01 _{2.3e-04}	6.62e-01 _{1.6e-05}
ZDT2	3.26e-01 _{3.4e-04}	3.29e-01 _{2.4e-05}
ZDT3	5.15e-01 _{1.3e-04}	5.16e-01 _{3.2e-05}
ZDT4	6.56e-01 _{3.3e-03}	6.62e-01 _{4.2e-05}
ZDT6	3.88e-01 _{1.6e-03}	4.01e-01 _{4.4e-05}
DTLZ1	6.65e-01 _{2.1e-01}	7.18e-01 _{4.7e-02}
DTLZ2	3.73e-01 _{5.1e-03}	3.76e-01 _{7.2e-03}
DTLZ3	0.00e+00 _{0.0e+00}	2.68e-01 _{1.1e-01}
DTLZ4	3.76e-01 _{6.4e-03}	3.72e-01 _{7.3e-03}
DTLZ5	9.28e-02 _{1.7e-04}	9.39e-02 _{2.4e-05}
DTLZ6	0.00e+00 _{0.0e+00}	9.49e-02 _{2.2e-05}
DTLZ7	2.71e-01 _{3.5e-03}	2.75e-01 _{3.3e-03}
UF1	5.03e-01 _{3.7e-02}	5.86e-01 _{2.3e-02}
UF2	6.06e-01 _{9.8e-03}	6.17e-01 _{6.7e-03}
UF3	3.14e-01 _{4.3e-02}	3.36e-01 _{5.5e-02}
UF4	2.47e-01 _{2.6e-03}	2.53e-01 _{2.8e-03}
UF5	5.12e-02 _{7.4e-02}	1.00e+00 _{0.0e+00}
UF6	1.80e-01 _{4.9e-02}	1.90e-02 _{3.6e-02}
UF7	3.04e-01 _{6.7e-02}	4.25e-01 _{7.2e-03}
UF8	1.44e-01 _{4.1e-02}	1.49e-01 _{1.9e-02}
UF9	2.90e-01 _{7.6e-02}	2.97e-01 _{3.8e-02}
UF10	1.90e-03 _{5.2e-03}	8.77e-02 _{3.8e-02}

(b)HV: Median and IQR

	VGSPSO	GSPSO
ZDT1	6.60e-01 _{3.4e-04}	6.62e-01 _{1.6e-05}
ZDT2	3.26e-01 _{6.4e-04}	3.29e-01 _{2.4e-05}
ZDT3	5.15e-01 _{1.7e-04}	5.16e-01 _{3.8e-05}
ZDT4	6.56e-01 _{6.6e-03}	6.62e-01 _{5.3e-05}
ZDT6	3.89e-01 _{2.0e-03}	4.01e-01 _{9.1e-05}
DTLZ1	7.40e-01 _{4.7e-02}	7.42e-01 _{1.6e-02}
DTLZ2	3.50e-01 _{8.8e-03}	3.56e-01 _{1.2e-02}
DTLZ3	0.00e+00 _{0.0e+00}	3.54e-01 _{2.4e-01}
DTLZ4	3.75e-01 _{8.6e-03}	3.92e-01 _{6.3e-03}
DTLZ5	9.28e-02 _{1.8e-04}	9.39e-02 _{4.6e-05}
DTLZ6	0.00e+00 _{0.0e+00}	9.49e-02 _{1.9e-05}
DTLZ7	2.81e-01 _{3.6e-03}	2.86e-01 _{4.8e-03}
UF1	5.20e-01 _{6.4e-02}	5.87e-01 _{3.5e-02}
UF2	6.09e-01 _{5.6e-03}	6.18e-01 _{9.7e-03}
UF3	2.93e-01 _{5.9e-02}	3.39e-01 _{8.5e-02}
UF4	2.47e-01 _{3.3e-03}	2.54e-01 _{3.3e-03}
UF5	0.00e+00 _{8.9e-02}	1.01e-01 _{0.0e+00}
UF6	1.84e-01 _{3.7e-02}	3.03e-01 _{2.3e-03}
UF7	2.89e-01 _{7.0e-02}	4.25e-01 _{1.5e-02}
UF8	1.49e-01 _{7.3e-02}	1.52e-01 _{3.3e-02}
UF9	3.03e-01 _{1.6e-01}	3.91e-01 _{3.7e-02}
UF10	0.00e+00 _{0.0e+00}	9.04e-02 _{5.2e-02}

TABLE VIII

(a) SPREAD: Mean and standard deviation

	VGSPSO	GSPSO
ZDT1	3.76e-01 _{2.7e-02}	6.66e-02 _{7.0e-03}
ZDT2	3.77e-01 _{2.1e-02}	6.66e-02 _{1.0e-02}
ZDT3	7.44e-01 _{1.2e-02}	7.07e-01 _{3.0e-03}
ZDT4	3.97e-01 _{2.6e-02}	9.36e-02 _{1.2e-02}
ZDT6	3.72e-01 _{2.9e-02}	6.40e-02 _{5.8e-01}
DTLZ1	8.78e-01 _{1.5e-01}	6.95e-01 _{3.3e-02}
DTLZ2	6.92e-01 _{2.7e-02}	6.41e-01 _{3.0e-02}
DTLZ3	1.06e+00 _{1.0e-01}	7.86e-01 _{2.7e-01}
DTLZ4	6.78e-01 _{4.1e-02}	6.68e-01 _{8.1e-02}
DTLZ5	4.49e-01 _{4.4e-02}	1.40e-01 _{4.7e-02}
DTLZ6	7.99e-01 _{5.1e-02}	1.46e-01 _{5.4e-02}
DTLZ7	7.82e-01 _{5.2e-02}	7.30e-01 _{2.7e-02}
UF1	1.50e+00 _{9.2e-02}	8.32e-01 _{5.8e-02}
UF2	6.21e-01 _{4.6e-02}	6.02e-01 _{6.9e-02}
UF3	1.06e+00 _{5.1e-02}	8.12e-01 _{9.8e-02}
UF4	5.11e-01 _{3.3e-02}	4.72e-01 _{5.8e-02}
UF5	1.23e+00 _{1.3e-01}	9.15e-01 _{7.4e-02}
UF6	1.17e+00 _{1.1e-01}	1.03e+00 _{1.8e-01}
UF7	1.07e+00 _{2.0e-01}	9.12e-01 _{7.1e-02}
UF8	8.18e-01 _{7.9e-02}	7.08e-01 _{4.8e-02}
UF9	8.42e-01 _{3.8e-02}	8.25e-01 _{7.0e-02}
UF10	7.67e-01 _{5.7e-02}	7.57e-01 _{1.1e-01}

(b) SPREAD: Median and IQR

	VGSPSO	GSPSO
ZDT1	$3.80e-01$ $2.8e-02$	$6.66e-02$ $1.3e-02$
ZDT2	$3.76e-01$ $2.8e-02$	$6.42e-02$ $2.0e-02$
ZDT3	$7.46e-01$ $1.5e-02$	$7.08e-01$ $2.8e-03$
ZDT4	$3.88e-01$ $2.4e-02$	$9.45e-02$ $1.4e-02$
ZDT6	$3.71e-01$ $4.7e-02$	$2.75e-01$ $1.2e+00$
DTLZ1	$8.34e-01$ $2.2e-01$	$6.92e-01$ $3.9e-02$
DTLZ2	$6.86e-01$ $2.7e-02$	$6.41e-01$ $2.7e-02$
DTLZ3	$1.02e+00$ $1.5e-01$	$6.53e-01$ $6.5e-02$
DTLZ4	$6.85e-01$ $2.0e-02$	$6.56e-01$ $1.1e-01$
DTLZ5	$4.51e-01$ $5.8e-02$	$1.26e-01$ $3.4e-02$
DTLZ6	$7.94e-01$ $5.7e-02$	$1.29e-01$ $1.8e-02$
DTLZ7	$7.95e-01$ $9.7e-02$	$7.27e-01$ $2.8e-02$
UF1	$1.46e+00$ $1.2e-01$	$8.41e-01$ $8.2e-02$
UF2	$6.15e-01$ $6.5e-02$	$5.84e-01$ $9.4e-02$
UF3	$1.05e+00$ $4.3e-02$	$8.29e-01$ $1.2e-01$
UF4	$5.07e-01$ $2.8e-02$	$4.65e-01$ $9.3e-02$
UF5	$1.20e+00$ $2.5e-01$	$9.00e-01$ $9.2e-02$
UF6	$1.13e+00$ $1.7e-01$	$9.49e-01$ $3.2e-01$
UF7	$1.14e+00$ $3.5e-01$	$9.10e-01$ $9.9e-02$
UF8	$8.19e-01$ $1.4e-01$	$6.96e-01$ $5.0e-02$
UF9	$8.47e-01$ $5.2e-02$	$7.66e-01$ $1.1e-01$
UF10	$7.67e-01$ $1.1e-01$	$7.56e-01$ $1.4e-01$

TABLE IX

(a) HV: Mean and standard deviation

	VGSPSO	GSPSO
DTLZ1	$5.86e-01$ $2.7e-01$	$7.38e-01$ $7.1e-03$
DTLZ2	$3.72e-01$ $5.7e-03$	$3.73e-01$ $5.3e-03$
DTLZ3	$0.00e+00$ $0.0e+00$	$3.16e-01$ $8.1e-02$
DTLZ4	$3.37e-01$ $1.1e-01$	$3.60e-01$ $4.9e-03$
DTLZ5	$9.28e-02$ $2.4e-04$	$9.38e-02$ $9.2e-05$
DTLZ6	$0.00e+00$ $0.0e+00$	$9.49e-02$ $2.0e-05$
DTLZ7	$2.80e-01$ $3.4e-03$	$2.87e-01$ $3.8e-03$

(b) HV: Median and IQR

	VGSPSO	GSPSO
DTLZ1	$7.21e-01$ $9.2e-02$	$7.38e-01$ $1.2e-02$
DTLZ2	$3.74e-01$ $9.4e-03$	$3.93e-01$ $6.4e-03$
DTLZ3	$0.00e+00$ $0.0e+00$	$3.52e-01$ $1.4e-02$
DTLZ4	$3.74e-01$ $4.0e-03$	$3.99e-01$ $1.0e-02$
DTLZ5	$9.28e-02$ $2.5e-04$	$9.38e-02$ $1.2e-04$
DTLZ6	$0.00e+00$ $0.0e+00$	$9.49e-02$ $2.4e-05$
DTLZ7	$2.80e-01$ $3.9e-03$	$2.97e-01$ $7.3e-03$

TABLE X

(a) SPREAD: Mean and standard deviation

	VGSPSO	GSPSO
DTLZ1	$8.74e-01$ $1.3e-01$	$6.71e-01$ $4.3e-02$
DTLZ2	$7.17e-01$ $5.2e-02$	$6.45e-01$ $4.5e-02$
DTLZ3	$1.10e+00$ $1.2e-01$	$7.78e-01$ $2.9e-01$
DTLZ4	$7.04e-01$ $1.0e-01$	$6.75e-01$ $6.1e-02$
DTLZ5	$4.31e-01$ $4.0e-02$	$1.68e-01$ $6.6e-02$
DTLZ6	$8.06e-01$ $4.8e-02$	$1.15e-01$ $1.4e-02$
DTLZ7	$7.45e-01$ $6.3e-02$	$6.88e-01$ $4.0e-02$

(b) SPREAD: Median and IQR

	VGSPSO	GSPSO
DTLZ1	$8.14e-01$ $1.8e-01$	$6.70e-01$ $7.3e-02$
DTLZ2	$7.11e-01$ $6.6e-02$	$6.47e-01$ $7.1e-02$
DTLZ3	$1.07e+00$ $2.2e-01$	$6.48e-01$ $5.0e-02$
DTLZ4	$6.80e-01$ $5.3e-02$	$6.70e-01$ $6.1e-02$
DTLZ5	$4.37e-01$ $6.8e-02$	$1.32e-01$ $6.6e-02$
DTLZ6	$7.86e-01$ $5.4e-02$	$1.15e-01$ $2.1e-02$
DTLZ7	$7.31e-01$ $1.2e-01$	$7.04e-01$ $6.1e-02$

The above analysis leads to a conclusion that GSPSO can attain improved overall performance compared to the classic algorithms for the majority of the experiments conducted. This improvement is primarily ascribed to the new evolution strategy exploiting the geometric structure of the scattered points on the optimal Pareto front. It is argued that these

scattered points can provide more helpful information than the dominated points. With the help of this information, a population can follow the normal direction to enter a new and better region during the evolution process. In this sense, this region is deterministic whereas the strategies employed by other algorithms only focus on random evolution without a deterministic idea.

B.1. Evaluation of Approximation and Convergence

To evaluate the performance of the proposed algorithm in terms of approximation and convergence, some standard test problems, namely ZDT2, DTLZ6 and UF8, are chosen from the MOO problem suites mentioned above to conduct further simulation experiments. These test problems are solved using five different algorithms with the same set of configuration parameters. Specifically, the population size is set to 100; the archive size is set to 100; and the maximum number of iteration is set to 25,000.

Fig. 13 through to Fig.15 show the true Pareto fronts and the optimal Pareto fronts obtained by using these five algorithms in the three test problems, respectively. The true Pareto front for each problem is shown in the first sub-figure highlighted using red color. It can be seen that the Pareto fronts obtained using GSPSO are the closest to the true Pareto fronts in all the three test problems. For ZDT2, all the algorithms can achieve good approximation to the true Pareto front. In this case, GSPSO is still better than others especially in the distribution of the second objective. In the case of DTLZ6, AbYSS and GSPSO can achieve optimal results that well approximate the true Pareto front, whereas those obtained by NSGAI and MOCeII are largely different from the true Pareto front. The true Pareto front of UF8 is a small part of a unit sphere and thus its shape is a surface. Fig. 15 shows that GSPSO has a relative uniform and large front compared to other algorithms whereas the Pareto fronts obtained using other methods are far different from the true Pareto front.

In this experiment, convergence speed is also analyzed particularly in terms of the running time and the trajectory of an indicator. The time cost of each algorithm obtained statistically in 50 times of simulation experiments is shown in Fig. 16, which includes the worse value, average value and best value. It can be seen from the figure that GSPSO and pMOEAD have outstanding performance in terms of efficiency. On the contrary, much higher time cost is incurred in MOCeII and NSGAI for the same number of function evaluations.

Table III and Table IV show the final results of the EPSILON indicator of the test problems. Since these results are used to evaluate the speed of convergence, the running time is not included for consideration. These algorithms have different speeds of evolution and different times of iteration when tested on different problems. Some of them can approximate the true Pareto front and arrive at the peak value with a small number of iteration whereas this process can be very slow in other algorithms. Fig. 17 through to Fig. 19 show the trajectories of EPSILON for the two dimensional problem ZDT2 and the three dimensional problems DTLZ6 and UF8, respectively. For ZDT2, GSPSO has the fastest convergence speed and attains best approximation of the true Pareto front

while NSGAI, MOCell and pMOEAD have similar routes of evolution. Some parts of the EPSILON curve are flat, which indicates that no preferable particles are produced during these iteration procedures. Moreover, AbYSS converges to the Pareto front with the slowest speed. For DTLZ6 and UF8, GSPSO has the fastest convergence speed while NSGAI, MOCell and pMOEAD have similar convergence speeds.

The above analysis indicates that GSPSO can solve the test problems with a very fast convergence speed and at a very low cost of running time. The deterministic normal direction can improve convergence speed to a large extent. Besides, the results also show that GSPSO has good robustness in terms of convergence speed.

Fig. 13. Optimal Pareto fronts obtained for ZDT2.

Fig. 14. Optimal Pareto fronts obtained for DTLZ6.

Fig. 15. Optimal Pareto fronts obtained for UF8.

Fig. 16. Time costs of the algorithms for different test problems.

Fig. 17. Change of EPSILON in ZDT2.

Fig. 18. Change of EPSILON in DTLZ6.

Fig. 19. Change of EPSILON in UF8.

B.2 Further statistical analysis of the experiment results

As evidenced in the analysis of experiment results, GSPSO can attain better performance than other algorithms in most cases. However, it should be ensured that the differences observed are statistically different. Therefore, the boxplots, t-test and the Wilcoxon test [31] are used to conduct further statistical analysis on the experiments results.

From Fig. 20 to Fig. 22 the boxplots of the HV indicator are shown for the ZDT2, DTLZ6 and UF8, respectively. The higher the box presents, then the better value HV is. In this sense, GSPSO has better mean values for these problems. It also can be seen that pMOEAD and AbYSS achieve bad results.

Furthermore, paired samples t-test method is adopted to evaluate the HV distributions obtained by these methods for ZDT2, DTLZ6 and UF8. The significance level $\alpha = 0.05$. For the comparisons between the algorithms and GSPSO the null hypothesis $H_0: m_A = m_B$, which means at the 0.05 level the mean value of sample A is not significantly different with sample B. The alternate hypothesis $H_1: m_A < m_B$. For the comparisons in which GSPSO is not involved, the alternate hypothesis H_1 is: $m_A \neq m_B$. The test results are shown in Table XI, XII and XIII. The symbol $\blacktriangle$ represents H_0 is rejected whereas ∇ is for acceptance. The results related to GSPSO show that H_0 is rejected, which means the GSPSO mean value is significantly more than other methods.

Alternatively, the Wilcoxon rank-sum test can be used to determine the significance of the obtained results, which is used to test whether two distributions are the same under certain conditions. In this experiment, the null hypothesis is that two distributions have the same median. A $\blacktriangle$ symbol implies $p\text{-value} < 0.5$, then the null hypothesis is rejected; otherwise, a ∇ is used to indicate that it is accepted. Table XIV and Table XV list the Wilcoxon test results for ZDT and UF, respectively. It can be seen from these two tables that the null hypothesis is rejected, which further proves that the median values in the boxplots are different.

The better distributions of the median values of GSPSO with respect to HV indicate that the optimal Pareto fronts obtained using GSPSO achieve very good uniformity and approximation, which is mainly ascribed to the new evolution strategy and the uniformity-ensuring operations.

Fig. 20. Boxplots of HV values ZDT2 .

Fig. 21. Boxplots of HV values for DTLZ6.

Fig. 22. Boxplots of HV values for UF8.

TABLE XI
PAIRED SAMPLE T-TEST FOR ZDT2

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▲	▲	▲	▲
pMOEAD		▲	▲	▲
MOCeII			▽	▲
AbYSS				▲

TABLE XII
PAIRED SAMPLE T-TEST FOR DTLZ6

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▲	▽	▲	▲
pMOEAD		▲	▲	▲
MOCeII			▽	▲
AbYSS				▲

TABLE XIII
PAIRED SAMPLE T-TEST FOR UF8

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▽	▲	▲	▲
pMOEAD		▲	▲	▲
MOCeII			▽	▲
AbYSS				▲

TABLE XIV
WILCOXON TEST FOR HV IN THE ZDT SUITE

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▲	▽	▲	▽
pMOEAD		▽	▲	▽
MOCeII			▲	▽
AbYSS				▽

TABLE XV
WILCOXON TEST FOR HV IN THE UF SUITE

UF1. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	—	▽	▽	▲
pMOEAD		▽	▽	▲
MOCeII			—	▲
AbYSS				▲

UF2. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▲	—	—	▲
pMOEAD		▽	▽	▲
MOCeII			—	▲
AbYSS				▲

UF3. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	—	▽	▽	▲
pMOEAD		▽	▽	▲
MOCeII			—	▲
AbYSS				▲

UF4. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	—	▲	▲	▲
pMOEAD		▲	▲	▲
MOCeII			—	▽
AbYSS				▽

UF5. HV.	SPEA2	MOEAD	AbYSS	GSPSO
NSGAII	—	▲	▲	▲
pMOEAD		▲	▲	▲
MOCeII			—	—
AbYSS				—

UF6. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	—	—	—	▲
pMOEAD		—	▲	▲
MOCeII			—	▲
AbYSS				▲

UF7. HV.	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAII	▲	▽	▽	▲
pMOEAD		▽	▽	—
MOCeII			—	▲
AbYSS				▲

UF8. HV.

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAI	▽	▽	▽	▽
pMOEAD		▲	▲	—
MOCeII			—	▽
AbYSS				—

UF9. HV.

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAI	▽	▽	▽	▲
pMOEAD		▽	▽	▲
MOCeII			—	▲
AbYSS				▲

UF10. HV.

	pMOEAD	MOCeII	AbYSS	GSPSO
NSGAI	▽	▲	▲	▲
pMOEAD		▲	▲	▲
MOCeII			—	▲
AbYSS				▽

C. Effectiveness Analysis for the Dimension Reduction Strategy

A feature of the proposed algorithm is that a dimension reduction strategy is adopted for optimization problems with more than three dimensions. To demonstrate its effectiveness, two groups of experiments are conducted with and without a true Pareto front.

C.1. Effectiveness Analysis with a True Pareto Front

To perform a quantitative evaluation of the effectiveness of the dimension reduction method, this experiment aims to study a series of DTLZ problems with three objectives by comparing their Pareto fronts after dimension reduction to the true Pareto fronts. The series of DTLZ problems are solved by reducing their numbers of dimensions to two. A relative error measure is proposed to evaluate the performance. The process of calculating relative error is as follows:

- 1) For each point on the Pareto front, after it is reduced to a two-dimensional problem, find out its design variables;
- 2) Evaluate the values of the three objective functions using the values of the current design variables and according to the problem definition;
- 3) Find the nearest point on the true Pareto front and calculate its Euclidian distance (marked as Dis_1 in Fig. 20);
- 4) Calculate the distance (marked as Dis_2 in Fig. 23) between the point on the true Pareto front and the original point;
- 5) Divide Dis_1 by Dis_2 . The result is the relative error for the current point on the Pareto front. The average of the relative errors of all the points is then the relative error between the Pareto front after dimension reduction and the true Pareto front.

In Fig. 23, point A is a point on the current Pareto front in the three-dimensional space; the red points are a subset on the true Pareto front; and point B is the closest point to point A. Dis_1 is the distance between A and B and Dis_2 is the distance between B and O. The ratio of Dis_1 to Dis_2 is then the relative error.

The relative errors for the series of DTLZ problems are shown in Table XVI, which are all quite small and acceptable especially in DTLZ3 and DTLZ5. The largest relative error of the dimension reduction is approximately 10% (0.1046).

Therefore, the Pareto front obtained using the proposed GSPSO algorithm with a dimension reduction strategy well approximates the true Pareto front.

Fig. 23. An example of relative error

TABLE XVI
THE RELATIVE ERRORS IN DTLZ PROBLEMS

Problem	DTLZ 1	DTLZ 2	DTLZ 3	DTLZ 4	DTLZ 5	DTLZ 6	DTLZ 7
Relative Error	0.0140	0.0145	0.0003	0.1046	0.0023	0.0056	0.0028

C.2. Effectiveness Analysis without a True Pareto Front

Another experiment is conducted on a series of DTLZ problems which have a larger number of objectives (nine objectives are selected in this case). The main focus is to evaluate the dimension reduction strategy for the cases in which a true Pareto front is unavailable. The configuration of the experiment is as follows: the population size is set to 100; the archive size is set to 1000; and the number of iteration must not exceed 4000. The numbers of evaluations for the other algorithms are shown in Table XVII. Because the true Pareto fronts are unknown, the Pareto fronts obtained using GSPSO will be compared to those obtained using the other three algorithms in two ways, namely using GSPSO as the basis of comparison and using another algorithm as the basis of comparison.

Dimension reduction is performed on the DTLZ problems with nine objectives, resulting in three clusters. The first one contains objectives 1, 3 and 4; the second one involves objectives 2 and 7; and the last one contains the remaining objectives. Table XVII displays the EPSILON values obtained when NSGAI, pMOEAD, MOCeII and GSPSO are used to solve the DTLZ problems. The EPSILON values obtained by GSPSO are used as the basis of comparison, that is, if EPSILON is larger than 1, it means GSPSO performs better. It can be concluded from the table that: (1) GSPSO obtains the best Pareto front; (2) NSGAI obtains the second best; and (3) pMOEAD is the worst in this case. For DTLZ2 with the same number of evaluations (i.e. 100,000), there is little difference between the different algorithms. The performances of these algorithms for DTLZ1 and DTLZ2 have large differences. This conclusion can be verified by further comparing the results obtained for DTLZ3, DTLZ4 and DTLZ5. In these cases, each algorithm is executed for 25,000 function evaluations. GSPSO clearly outperforms the other algorithms for DTLZ3. However, NSGAI has the best convergence speed for DTLZ4 and DTLZ5. The main

difference between DTLZ2 and DTLZ3 is in that the numbers of evaluation are different. In these cases, the performances of these algorithms are to a large extent different, which indicates that the number of evaluation (shown in the last column) is also an important factor to consider when comparing the results.

TABLE XVII

EPSILON COMPARISON - THREE OBJECTIVES AFTER REDUCTION
(GSPSO IS USED AS THE BASIS OF COMPARISON)

Problems	NSGAI	pMOEAD	MOCcell	Evals
DTLZ1	13.7258	23.4068	21.7916	100,000
DTLZ2	0.7683	0.4422	0.5748	100,000
DTLZ3	23.9535	195.5757	76.8350	25,000
DTLZ4	0.8102	0.5387	0.5695	25,000
DTLZ5	0.1016	0.2505	0.1055	25,000
DTLZ6	0.0437	0.6460	0.0739	25,000
DTLZ7	0.1156	0.0857	0.1213	25,000

When the problems are reduced to two dimensions in GSPSO, two clusters are produced: the first one contains objectives 1, 3, 4 and 7 and the second one contains the remaining objectives. Table XVIII shows a comparison of the EPSILON values obtained in different algorithms when GSPSO is used as the basis of comparison. Compared with the results in Fig. 16, the order of deviation magnitudes is 10^{-3} or 10^{-4} . The distribution of the obtained results is roughly consistent with Table XVII, which shows that the dimension reduction strategy has good robustness.

TABLE XVIII

EPSILON COMPARISON - TWO OBJECTIVES AFTER REDUCTION
(GSPSO IS USED AS THE BASIS OF COMPARISON)

Problems	NSGAI	pMOEAD	MOCcell	Evals
DTLZ1	13.725	23.406	4.9657	100,000
DTLZ2	0.7254	0.3884	0.2915	100,000
DTLZ3	23.953	195.57	42.402	25,000
DTLZ4	0.7906	0.4515	0.5285	25,000
DTLZ5	0.1016	0.2508	0.1799	25,000
DTLZ6	0.0428	0.6452	0.0731	25,000
DTLZ7	0.1149	0.0852	0.01225	25,000

Tables XIX and XX show comparisons of the EPSILON values when the total number of objectives is reduced to three and two, respectively. In this case, another algorithm is used as the basis of comparison. None of the values in the table is large, which indicates that GSPSO is not worse than the other three algorithms (in this case if EPSILON is less than 1, GSPSO performs better). These test cases further confirm the effectiveness of the dimension reduction strategy.

TABLE XIX

EPSILON COMPARISON - THREE OBJECTIVES AFTER REDUCTION
(GSPSO IS NOT THE BASIS OF COMPARISON)

Problems	NSGAI	pMOEAD	MOCcell	Evals
DTLZ1	0.0872	0.1164	0.1239	100,000
DTLZ2	0.5821	0.5815	0.5853	100,000
DTLZ3	0.7547	0.4059	0.7626	25,000
DTLZ4	0.6823	0.6762	0.6837	25,000
DTLZ5	0.1777	0.1895	0.2235	25,000
DTLZ6	0.8812	0.0015	0.7996	25,000
DTLZ7	1.2646	0.1203	0.0781	25,000

TABLE XX

EPSILON COMPARISON - TWO OBJECTIVES AFTER REDUCTION
(GSPSO IS NOT THE BASIS OF COMPARISON)

Problems	NSGAI	pMOEAD	MOCcell	Evals
DTLZ1	0.0417	0.0641	0.1000	100,000
DTLZ2	0.7069	0.7066	0.7076	100,000
DTLZ3	0.7093	0.6142	0.7093	25,000

DTLZ4	0.7069	0.7069	0.7075	25,000
DTLZ5	0.1779	0.1893	0.2235	25,000
DTLZ6	0.8819	0.0018	0.7914	25,000
DTLZ7	1.2648	0.1215	0.0792	25,000

The experiments introduced in this section demonstrate the effectiveness of the dimension reduction strategy. It further confirms that this strategy is an important part of the GSPSO algorithm. Such a strategy ensures that the problems with a large number of objectives can be solved efficiently. These experiments also conclude that the strategy has a satisfactory performance in terms of robustness.

D. Experiment of Electric Vehicle Design Optimization

In addition to the classic testing problems, the proposed algorithm has also been applied to an optimization problem in an electric vehicle design [32] to further demonstrate its effectiveness as well as to explain how it is applied to real engineering problems.

The optimization model of this problem contains several elements. Specifically, the design variables include the main reducer ratio and the two gear ratios in transmission, which is represented as a vector $\mathbf{x} = [ig_0, ig_1, ig_2]$. Additionally, some other relevant parameters have also been considered, as listed in Table XXI.

TABLE XXI

PARAMETERS OF AN ELECTRIC VEHICLE

Parameter name	Value	Comment
t_0	0.2	Shift gear time
u_0	30	Shift gear speed
s_0	80	Maximum speed
T_{max}	170	Maximum torque
P_{max}	27	Maximum power
g	9.8	Constant of gravity
m	1150	Mass
f	0.2	Friction factor
C_d	0.33	Drag factor
A	2.16	Drag square
σ	1.1	Rotating mass factor
r	0.3075	Tire radius
η_{DOD}	0.8	Discharge depth
η_{mc}	0.8	Controller efficiency
U	3.2	Voltage of one battery
N	98	Number of battery
L	110	Distance for one working condition

Moreover, three optimization objectives are considered, including the acceleration time T from 0 to the maximum speed s_0 , gradeability G and driving distance D for a given amount of energy. Calculation of the main objectives and constraints is given below.

The acceleration time and gradeability objectives are calculated using the Equations (19) and (20), respectively.

$$T = \frac{1}{3.6} \int_0^{s_0} \frac{\delta m}{F_t - (Gf + \frac{C_d A}{21.15} u^2)} du \quad (19)$$

$$G = \arcsin \frac{F_t - F_w}{mg\sqrt{1+f^2}} - \arctan f \quad (20)$$

The driving distance for one working condition mainly consists of two factors, namely energy cost in the constant speed phase and energy cost in the acceleration phase. The required power for a given constant speed u_a is:

$$P = \frac{u_a}{3600\eta_i} (mgf + \frac{C_d A u_a^2}{21.15}) \quad (21)$$

The motor rotation speed n at the speed of u_a is:

$$n = \frac{u_a i g_0 i g_i}{0.377 r} \quad (22)$$

The motor Torque at the speed of u_a is:

$$Torque = \frac{r}{\eta_i i g_0 i g_i} (mgf + \frac{C_d A u_a^2}{21.15}) \quad (23)$$

The motor efficiency can then be calculated by conducting an interpolation operation using n and Torque, as shown in Equation (24).

$$\eta_e = \eta(n, Torque) \quad (24)$$

Therefore, the energy cost for a constant speed is:

$$W_1 = P * t / \eta_e \quad (25)$$

The acceleration phase can be regarded as a large number of constant phases and the calculation process of w_2 is similar to that for the constant speed phase. The total energy of the battery is then:

$$W_t = CUN\eta_{DOD} \quad (26)$$

In this study, the CYC_ECE_EUDC_LOW working condition is adopted, and the total driving distance for this working condition and amount of energy can be calculated using Equation (27) where L is the driving distance for one working condition.

$$D = \frac{w_t}{w_1 + w_2} L \quad (27)$$

In this mathematical model, the minimum value needs to be found for the first objective while the maximum values need to be found for the other objectives. All the three objectives are given in a unified form as shown in Equation (28).

$$\begin{aligned} F_1 &= T \\ F_2 &= \frac{1}{1+G} \\ F_3 &= \frac{100}{D} \end{aligned} \quad (28)$$

The experiment results given in Fig. 24 indicate that the optimal Pareto fronts obtained by different algorithms are pretty similar. Specifically, the fifth subplot is the optimal Pareto front obtained using GSPSO while the sixth subplot shows the result when the dimension reduction strategy is used in GSPSO. As shown in the figure, GSPSO is reasonably better than NSGAI and MOCcell while AbYSS and pMOEAD achieve the worst performance. The fifth subplot and the sixth subplot confirm the effectiveness of the

dimension reduction strategy. This experiment demonstrates that the evolution strategy based on the geometric structure of the Pareto front can also be used to solve engineering optimization problems accurately and effectively. Besides, the result shown in the fifth subplot is better than that in the sixth subplot because information lost in this case is less than the case where the number of objectives is reduced to two.

Fig. 24. Optimal Pareto fronts obtained by different algorithms.

VIII. CONCLUSIONS AND FUTURE WORK

In this research, a novel evolution strategy is proposed for solving MOO problems, which improves effectiveness and efficiency of searching by exploiting the geometric structure of the Pareto front. On this basis, a local optimization process is developed in which a set of guiding points is obtained using the geometric structure represented as a parametric equation to guide particles' flights. In this way, the entire population can move towards a better region in each time of iteration so that the evolution process can quickly converge. In the early evolution phase, a population largely depends on the guiding points set whose influence will gradually decrease as the evolution process approaches the end. The main contribution of this work is summarized as follows:

- 1) A new algorithm based on the geometric structure of the Pareto front is proposed for MOO problems. Geometric information such as the normal direction is utilized to support the particles in moving to better regions.
- 2) A method based on trust region is proposed to improve uniformity of a population, which can insert points into the Pareto front according to its density distribution. Engineering designers can easily obtain some solutions from an arbitrary position in the feasible region. It is also possible to find interesting solutions from the points outside the geometric parameter space.
- 3) A dimension reduction strategy is developed based fuzzy correlation analysis, which can effectively reduce the number of dimensions for an optimization problem. Employing this strategy, the GSPSO algorithm can be used to solve the MOO problems with more than three objectives.

Through conducting several computational experiments and running a series of analysis on the simulation results, it can be concluded that GSPSO can outstanding performances

in terms of the quality of the final optimal Pareto front, speed of convergence and robustness. There are two areas where future work can be done to improve the GSPSO algorithm. The first is in the development of new mathematical models for handling the problems with more than three objectives without the need of conducting dimension reduction. The second involves exploring better fitness methods and better local optimization strategies.

Acknowledgements

The authors appreciate the financial supports from the National Science Foundation of China (Grant Nos. 61173126 and 61572427) and the 863 high-technology project of China (Grant No. 2011AA100804).

REFERENCES

- [1] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithms: NSGA-II," *IEEE Trans Evol Comput*, vol. 40, no. 8, pp. 182-97, 2002.
- [2] J. Knowles and D. Corne, "The Pareto archived evolution strategy: a new baseline algorithm for Pareto multiobjective optimization," In: *Proceedings of the 1999 congress on evolutionary computation-CEC 99*, 6-9 July 1999. Washington, DC, USA: IEEE.
- [3] J. Kennedy and R. Eberhart, "Particle swarm optimization," In *Proceedings of IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
- [4] Beausoleil, "Multiobjective scatter search applied to non-linear multiple criteria optimization," *European Journal of Operational Research*, vol. 169, no. 2, pp. 426-449, 2005.
- [5] X. Chen and Y. Li, "A modified PSO structure resulting in high exploration ability with convergence guaranteed," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37 no. 5, pp. 1271-1289, 2007.
- [6] Christopher, K.M, and Seppi., 2004, "The Kalman swarm: a new approach to particle motion in swarm optimization," In : *Proceedings Genetic and Evolutionary Computation Conference (GECCO)*, 26-30 June, Seattle, WA, 2004. Lecture Notes in Computer Science, vol. 3103, Berlin: Springer, pp. 140-150.
- [7] Clerc, M., "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," In: *Proceedings of the IEEE congress on evolutionary computation*. 1999, vol. 3, pp. 1951-1957.
- [8] M.. Reyes Sierra and C. A. Coello Coell, "Improving PSO-Based Multi-objective Optimization Using Croding, Mutation," In *Evolutionary Multi-Criterion Optimization (EMO 2005)*, LNCS 3410, pp. 505-519, 2005.
- [9] Mostaghim, S. and Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," In: *Proceedings of the IEEE swarm intelligence symposium*, Indianapolis, April, 2003, IN. Piscataway, NJ: IEEE Service Center, pp. 26-33.
- [10] Fieldsend, J.E. and Singh, "A multi-objective algorithm based upon particle swarm optimization and efficient data structure and turbulence," In : *Workshop on computational intelligence*, 2-4 School of Computer Science, University of Birmingham, UK, September, 2002, pp. 37-44.
- [11]Yen, G.G., and Leong, "Dynamic multiple swarms in multi-objective particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 39, no. 4, pp. 890-911, 2009.
- [12] Moubayed N.Al, Petrovski and J. McCall , "D2MOPSO: MOPSO Based on Decomposition and Dominance with Archiving Using Crowding Distance in Objective and Solution Spaces," *Evolutionary Computation*, vol. 22 no. 1, pp. 47-77 2014.
- [13] S. Z. Zhao and P. N. Suganthan, "Two-bests based multi-objective particle swarm optimizer," *Engineering Optimization*, vol. 43, no.1, pp.1-17, 2011.
- [14] S. K. Fan and J. M Chang, "A parallel particle swarm optimization algorithm for multi-objective optimization problems," *Engineering Optimization*, vol. 41, no. 7, pp. 673-697, 2009.
- [15] C. John., Steuben and Cameron, "Robust Optimization of Mixed-Integer Problems Using NURBS-Based Meta models," *Journal of Computing and Information Science in Engineering*, vol. 12, 2012.
- [16] L. Piegl, "On NURBS: A survey," *IEEE Computer Graphics and Applications*, vol. 11, no. 1, pp. 55-71, 1991.
- [17] B. Starly, A. Lau, and W. Sun, "Direct slicing of STEP based NURBS models for layered manufacturing," *Computer-Aided Design*, vol. 37, pp. 387-397, 20005.
- [18] Powell, M.J, and Y. Yuan, "A trust region algorithm for equality constrained optimization," *Math Prog*, vol.49, pp. 189-211, 1991.
- [19] P. Cignoni, and C. Montan, "DeWall: A fast divide and conquer Delaunay triangulation algorithm," *Computer-Aided Design*, vol. 30, no. 5, pp. 333-341, 1997.
- [20] S. Floater, and K Hormann, "Surface parameterization: A tutorial and survey," *Advances in multiresolution for geometric modeling mathematics and visualization*, pp. 157-186, 2005.
- [21] Sch. of Electr. Eng., Seoul Nat. Univ, "Constructing NURBS surface model from scattered and unorganized range data," *IEEE 3-D Digital Imaging and Modeling Proceedings*, pp. 312-320, 1999.
- [22] D. DeMers, and G.W.Cottrell, "Nonlinear dimensionality reduction," In *Advances in Neural Information Processing Systems 5*, D. Hanson, J. Cowan , L. Giles, Eds, 1993, pp. 580-587.
- [23] S. T. Roweis, and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 2323-2326, 2000.
- [24] J.B. Tenenbaum, and V. de Silva, "A global geometric framework for nonlinear dimensionality reduction," *Science*, pp. 2319-2323, 2000.
- [25] R. Malgady, and K. David, "Understanding Correlation Coefficients and Regression," *Journal of the American Physical Therapy Association*, vol. 66, pp. 110-120, 1986.
- [26] J. Zurovac, R. Brown, "Orthogonal design: A powerful method for comparative effectiveness research with multiple interventions," *Health Care Effectiveness*, 2014.
- [27] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D Uniform Design: A New Version of MOEA/D for Optimization Problems with Many Objectives," *Computers & Operations Research*, vol. 40, pp. 1648-1660, 2013.
- [28] K. Deb, A. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multiobjective optimization," *Evolutionary Multiobjective Optimization, Theoretical Advances and Applications*, pp. 105-145, 2001.
- [29] Q. Zhang, A. Zhou, S. Z. Zhao and P. N. Suganthan, "Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition," Technical Report CES-887, University of Essex and Nanyang Technological University, 2008.
- [30] J.Knowles, L. Thiele, and E. Zitzler, "A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers," Technical Report 214, Computer E ngineering and Networks Laboratory (TIK), ETH Zurich, 2006.
- [31] J. Derrac, S. Garcia, D. Molina, F. Herrera, "A Practical Tutorial on the Use of Nonparametric Statistical Tests as a Methodology for Comparing Evolutionary and Swarm Intelligence Algorithms," *Swarm and Evolutionary Computation*, Vol. 1, No. 1, pp. 3-18, 2011.
- [32] M. Ehsani, Y. M. Gao and A. Emadi, "Modern Electric, Hybrid Electric, and Fuel Cell Vehicles, Fundamentals, Theory, and Design", China Machine Press, Second Edition.