IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 28, 2019, accepted June 22, 2019, date of publication July 3, 2019, date of current version August 9, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2926584*

# Multi-Objective Particle Swarm Optimization Based on Fuzzy Optimality

**YONGPENG SHEN AND GAORUI GE**
College of Electrical and Information Engineering, Zhengzhou University of Light Industry, Henan 450002, China

Corresponding author: Yongpeng Shen (shenyongpeng@zzuli.edu.cn)

This work was supported in part by the Science and Technology Open Cooperation Project of Henan Province under Grant 182106000032, and in part by the National Natural Science Foundation of China under Grant 61803345.

**ABSTRACT** In order to overcome the limitations of the Pareto optimality in solving multi-objective optimization problems, a new optimality definition, fuzzy optimality is proposed, which considered both of the numbers of improved objectives and the extent of the improvements. Then, the fuzzy optimality-based multi-objective particle swarm optimization algorithm is presented. It inherits the basic structure of the particle swarm optimization and evaluates the particles by the fuzzy optimality. The numerical experiments are carried out on 6 representative test functions, and the results show that the proposed fuzzy optimality based multi-objective particle swarm optimization algorithm shows better performance on aspects of quality of solutions, robustness, and computational complexity, compared with the results of the NSGA-II and MOPSO. Finally, the efficacy and practicality of the proposed approach are validated in the APU fuel consumption and emissions multi-objective optimization problem.

**INDEX TERMS** Multi-objective, optimization, fuzzy optimality, particle swarm optimization.

## I. INTRODUCTION

Many real-world applications involve simultaneous optimization of multiple objectives which are often conflicting with each other and subject to a number of equality or inequality constraints, such as the motor controller tuning problem [1], the set-point tracking controller design problem [2], the hydraulic energy storage system [3], the power allocation problem of the multibeam satellite systems [4] and the test suite optimization problem [5]. Such problems are formulated as multi-objective optimization problems (MOPs) by researchers, in which the goal is to minimize or maximize the conflicting objectives simultaneously.

Mtaheuristic evolutionary algorithms have received much attention by researchers in the past as they have the ability to solve real world complex problems [6]. To solve the MOPs, various metaheuristic evolutionary approaches were proposed, such as the Vortex Multi-Objective Particle Swarm Optimization(MOVPSO) [7], cross-entropy based MOP [8], CMO(cooperative multi-Objective optimization) [9], MOPSO (multi-objective particle swarm optimization) [10] and the latest proposed MOWCA (multi-objective water cycle algorithm) [11], IMADE (immune multi-objective optimization algorithm with differential evolution inspired recombination) [12], DMMO (double-module immune algorithm) [13], MAP (memory-based adaptive partitioning algorithm) [14] and SAMOHS (self-adaptive multi-objective harmony search algorithm) [15].

One of the general characters of the aforementioned approaches is that they are all Pareto optimality (PO) based approaches and aimed at converging toward the Pareto front (PF), which is a set of optimal solutions in the Pareto sense. For the PO based approaches, the number of improved objectives, the extent of such improvements and the preference information of each objective are not taken into account in the calculation process. However, these issues are crucial in practice and usually only one 'compromise' solution is finally desired. Consequently, an additional multi-criteria decision making method is needed [16].

In this paper, a new optimality definition: *fuzzy optimality* (FO) is presented and the FO based multi-objective particle swarm optimization (FMOPSO) is proposed. For the proposed FO, the number of improved objectives and the extent of the improvements are both considered, the linguistic preference information of objectives are represented by the fuzzy

The associate editor coordinating the review of this manuscript and approving it for publication was Dong Wang.

triangular numbers and processed by fuzzy mathematics, thus it is much closer to the optimality judgement in human decision-making activities.

The rest of this paper is structured as follows. In section 2, several optimality concepts are reviewed and their basic features are analyzed. In section 3, FO is present and its calculation process is described in detail. Section 4 describes the critical steps and the overall process of the FMOPSO algorithm. In section 5, the FMOPSO is validated using several standard test functions. In section 6, the FMOPSO is further validated in the auxiliary power unit(APU) fuel consumption and emissions multi-objective optimizaiton problem. Finally, the conclusion is stated in section 6.

## II. STATE OF THE ART

In general, a constrained multi-objective optimization problem can be formulated in mathematical terms as follows,

$$\begin{aligned} &\textit{minimize} : [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})] \\ &\textit{subject to} : g_i(\boldsymbol{x}) \leq 0 \quad i = 1, 2, \cdots, m \end{aligned} \tag{1}$$

where $k$ is the number of objective functions,$f_i : \mathbb{R}^s \to \mathbb{R}$, $\boldsymbol{x} = [x_1, x_2, \cdots, x_s]^T$ is the vector of independent variables. $m$ is the number of inequality constraints.

For the MOPs, the most frequently used methods are PO based approaches. The aim of PO based multi-objective optimization approaches is to determine a particular set of values $\boldsymbol{x}^* = [x_1^*, x_1^*, \cdots, x_s^*]$ from the feasible solution space. Here, these $\boldsymbol{x}^*$ solutions are named as nondominated, noninferior or Pareto optimal solutions.

When PO is considered, it means that solutions cannot be improved in any of the objectives without degrading at least one of the other objectives. In mathematical terms, solution $\boldsymbol{x}^1$ is said to *Pareto dominates* $\boldsymbol{x}^2$ (also written as $\boldsymbol{x}^1 \succ \boldsymbol{x}^2$) when it meets the following conditions,

1. $f_i(\boldsymbol{x}^1) \leq f_i(\boldsymbol{x}^2), \forall\, i \in (1, 2, \cdots, k)$;
2. $\exists\, j \in (1, 2, \cdots, k)$: $f_j(\boldsymbol{x}^1) < f_j(\boldsymbol{x}^2)$.

solution $\boldsymbol{x}^1$ is called *Pareto optimal*, if there does not exist another solution that dominates it. However, the Pareto optimal always gives not a single solution, but rather a set of solutions called the *Pareto-optimal set* (Pareto set). The individual solution included in the Pareto-optimal set is called non-dominated solution. The image of the Pareto-optimal set under the objective functions is called *Pareto front*.

The definition of PO is ineffective to evaluate the independent variables objectively due to essentially three reasons,

1. Number of improved or equal objectives is not taken into account;
2. The extent of such improvements are not taken into account;
3. Preference information of different objectives are not taken into account.

Such issues are implicitly included in the common sense of optimality and are the essential elements in the human evaluation process. In order to overcome the limitations of the PO, a new definition-FO is introduced in next section.

Several other optimality concepts have been proposed in the literature. Through progressively expand the dominance area of solutions as the problem dimensionality increases, and thereby maintaining a relatively consistent level of solution discriminability across a range of many objectives, Zhu *et al.* proposed a generalized Pareto-optimality (GPO) to tackle the scalability problem of the PO based approaches [17]. He *et al.* reformulate the original large-scale multi-objective optimization problem into a low-dimensional single-objective optimization problem via problem reformulation, and proposed a large-scale multi-objective optimization framework (LSMOF) [18]. Through training the trade-off solutions to a neuro-fuzzy system (NFS), Das and Pratihar proposed a novel approach for neuro-fuzzy system-based multi-objective optimization to capture inherent fuzziness in engineering processes [19].

Alanis *et al.* assisted dynamic programming optimization framework and validated that it is capable of circumventing the increased complexity of the Pareto optimality [20]. Farina and Amato [21] proposed the $k$-optimality and $k_F$ optimality concepts. Each of these optimality is a sound extension both of the previous one and of Pareto-optimality. For these optimality definitions, the fuzzy membership is utilized as a tool for the numerical formalization and treatment of the size of improvements in the dominance definition. However, the preference information is not considered in these optimality definitions.

## III. FUZZY OPTIMALITY

In this section, FO is introduced. If the FO is considered in an MOP, the variable which gives smallest FO value is recognized as the best solution.

### A. BASIC CONCEPTS

To make the FO easier to be understood, the objective values of the candidate solutions set $IV$ are concisely expressed in matrix format as,

$$\boldsymbol{F}(IV) = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1k} \\ f_{21} & f_{22} & \cdots & f_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ f_{S1} & f_{S2} & \cdots & f_{Sk} \end{bmatrix} \tag{2}$$

where, as defined above, $k$ is the number of the objective functions, and $S$ is the number of the candidate independent variables, here $S$ is the rows of the matrix $IV$. $f_{ij}$ and $w_i$ are all crisp numbers, where $i \in (1, 2, \cdots, S), j \in (1, 2, \cdots, k)$.

In many circumstances, crisp numbers are inadequate to model real-life situations, since human judgements including preferences are often uncertain, vague and cannot be estimated by conventional quantitative terms. To resolve the vagueness and ambiguity of human decision-making activities, the fuzzy sets theory which was proposed by Zadeh is a natural candidate due to its capability of processing the data in the forms of fuzzy numbers [22].

The basic definitions of fuzzy set theory and FO are described as follows,

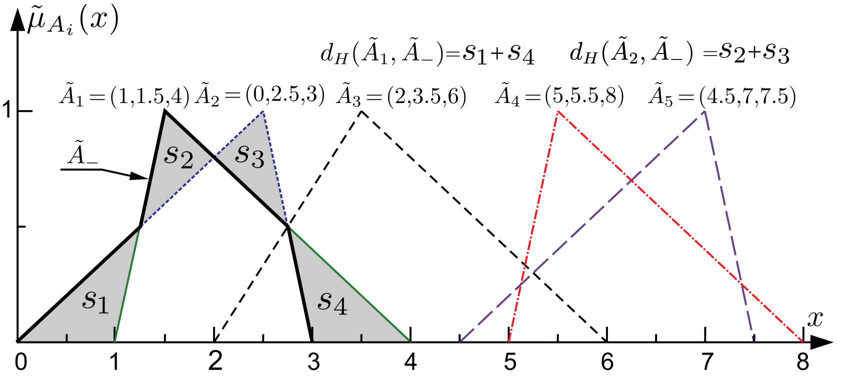


**FIGURE 1.** Schematic of five triangular fuzzy numbers, minimum fuzzy subset $\tilde{A}_-$ and Hamming distance between two fuzzy subsets.

*Definition 1:* A fuzzy subset $\tilde{A}$ in a universe of discourse $X$ is defined by a membership function $\mu_{\tilde{A}}(x) : X \to [0, 1]$, where $\mu_{\tilde{A}}(x)$ indicates the degree to which $x$ belongs to $\tilde{A}$ [22]. The value 0 means that $x$ is not a member of the fuzzy subset; the value 1 means that $x$ is fully a member of the fuzzy subset. The values between 0 and 1 characterize fuzzy members, which belong to the fuzzy subset only partially. An example about the fuzzy subset and membership function is given in Appendix A.

*Definition 2:* A triangular fuzzy number $\tilde{M}$ is a normal, convex fuzzy subset of $X$, with a piecewise linear membership function $\mu_{\tilde{M}}(x)$ defined by

$$\mu_{\tilde{M}}(x) = \begin{cases} (x-l)/(m-l) & l < x \leq m \\ (r-x)/(r-m) & m < x \leq r \\ 0 & otherwise \end{cases} \tag{3}$$

which is specified by the triplet of $\tilde{M} = (l, m, r)$, where $l$ and $r$ are the lower and upper bounds of the fuzzy number $\tilde{M}$ respectively, and $m$ is the modal value. An example about the triangular fuzzy number is given in Appendix A. Furthermore, the diagram of five triangular fuzzy numbers $(\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_5)$ are shown in Fig. 1.

*Definition 3:* $\tilde{A}_- = \widetilde{min}(\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_n)$ is the minimum fuzzy subset among $\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_n$, the member function $\mu_{\tilde{A}_-}(x)$ is defined as,

$$\mu_{\tilde{A}_-}(x) = \sup_{x=x_1\wedge\cdots\wedge x_n} \min\{\mu_{\tilde{A}_1}(x_1), \cdots, \mu_{\tilde{A}_n}(x_n)\} \tag{4}$$

where $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^n$, $\wedge$ is the fuzzy minimize operator. The membership function of the minimum fuzzy subset $\tilde{A}_-$ among $(\tilde{A}_1, \tilde{A}_2, \cdots, \tilde{A}_5)$ is shown in Fig. 1 by black thick lines. Another example about the minimum fuzzy subset is given in Appendix A.

*Definition 4:* $d_H(\tilde{A}_i, \tilde{A}_j)$ is the Hamming distance between fuzzy subset $\tilde{A}_i$ and $\tilde{A}_j$,

$$d_H(\tilde{A}_i, \tilde{A}_j) = \int_{x\in\mathbb{R}} |\mu_{\tilde{A}_i}(x) - \mu_{\tilde{A}_j}(x)|\, dx \tag{5}$$

As an example, the Hamming distance $d_H(\tilde{A}_1, \tilde{A}_-)$ and $d_H(\tilde{A}_2, \tilde{A}_-)$ are graphically demonstrated in Fig. 1, where $s_1, s_2, s_3, s_4$ are the area of the shaded areas which constitute the geometric metric of Hamming distance. As shown in Fig. 1, $d_H(\tilde{A}_1, \tilde{A}_-) = s_1 + s_4$, $d_H(\tilde{A}_2, \tilde{A}_-) = s_2 + s_3$. Another example about the Hamming distance between two fuzzy subset is given in Appendix A.

*Definition 5:* For $i * j$ triangular fuzzy numbers $\tilde{f}_{ij} = (l_{ij}, m_{ij}, r_{ij})$, $i \in (1, 2, \ldots, S)$, $j \in (1, 2, \ldots, k)$. Let $(\cdot)_j^{\max} = \max_{i=1,2,\ldots,S}\{(\cdot)_{ij}\}$, then the normalized fuzzy triangular number of $\tilde{f}_{ij}$ is described as,

$$\tilde{f}_{\Delta ij} = (\frac{l_{ij}}{r_j^{\max}}, \frac{m_{ij}}{m_j^{\max}}, \frac{r_{ij}}{l_j^{\max}} \wedge 1) \tag{6}$$

An example about the calculation of the normalized fuzzy triangular number is given in Appendix A.

*Definition 6:* For two triangular fuzzy numbers, $\tilde{x} = (l_x, m_x, r_x)$ and $\tilde{y} = (l_y, m_y, r_y)$, $\tilde{x} \cdot \tilde{y}$ is defined as,

$$\tilde{z} = \tilde{x} \cdot \tilde{y} = (l_x l_y, m_x m_y, r_x r_y) \tag{7}$$

Based on the basic concepts introduced above, each objective function value $f_{ij}$ is converted to triangular fuzzy number $\tilde{f}_{ij} = (f_{ij}, f_{ij}, f_{ij})$ and the objective values of the candidate solutions set $IV$ described by equation (2) is rewritten as

$$\tilde{\boldsymbol{F}}(IV) = \begin{bmatrix} \tilde{f}_{11} & \tilde{f}_{12} & \cdots & \tilde{f}_{1k} \\ \tilde{f}_{21} & \tilde{f}_{22} & \cdots & \tilde{f}_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{f}_{S1} & \tilde{f}_{S2} & \cdots & \tilde{f}_{Sk} \end{bmatrix} \tag{8}$$

Since human preferences are often vague and cannot estimate by an exact numerical value. In practice, the preferences information are often described in natural language, such as ‘very important’, ’fairly important’, ‘important’, ‘somewhat important’, ‘general’ and ‘not important’. In the proposed FO, the linguistic preference information is mapped to fuzzy triangular number and then processed by the aforementioned fuzzy operators. As an example, the linguistic preference information between three objectives {‘important’, ‘somewhat important’, ‘not important’} can be mapped to the fuzzy fuzzy triangular number set {(0.5, 0.5, 0.6), (0.3, 0.4, 0.5), (0, 0.1, 0.1)}. Here, the preference information are given in the format of triangular fuzzy numbers directly, and the preference vector is written as $\tilde{w} = [\tilde{w}_1, \tilde{w}_2, \cdots \tilde{w}_k]$.

### B. COMPUTATIONAL FLOW OF THE FO

For the candidate independent variable set $IV$ to be evaluated, the FO computational flow of each independent variable is summarized as the following steps,

*Step 1:* Map the linguistic preference information to fuzzy triangular preference vector $\tilde{w} = [\tilde{w}_1, \tilde{w}_2, \cdots \tilde{w}_k]$.

*Step 2:* Normalize the objective values $\tilde{\boldsymbol{F}}(IV)$ by column with equation (6).

*Step 3:* Obtain the fuzzy utility matrix by the following equation,

$$\tilde{r}_{ij} = \tilde{w}_j \cdot \tilde{f}_{\Delta ij}, \quad \forall\, i, j \tag{9}$$

where $i \in (1, 2, \cdots, S), j \in (1, 2, \cdots, k)$, the '$\cdot$' operator is defined by definition 6.

*Step 4:* Obtain the fuzzy ideal solution $\tilde{M}^*$,

$$\tilde{M}^* = (\tilde{M}_{1-}, \tilde{M}_{2-}, \cdots \tilde{M}_{k-}) \tag{10}$$

where $\tilde{M}_{j-} = \widetilde{min}(\tilde{r}_{1j}, \cdots \tilde{r}_{Sj}), j \in (1, 2, \cdots, k)$, the membership function of $\tilde{M}_{j-}$ is described by the following equation,

$$\mu_{\tilde{M}_{j-}}(r) = \sup_{r=r_1\wedge\cdots\wedge r_S} \min\{\mu_{\tilde{r}_{1j}}(r_1), \cdots, \mu_{\tilde{r}_{Sj}}(r_S)\} \tag{11}$$

where $(r_1, r_2, \cdots, r_S) \in \mathbb{R}^S$.

*Step 5:* Calculate the Hamming distance $D_i$ between the $i$th solution and the fuzzy ideal solution $\tilde{M}^*$,

$$D_i = \sqrt{\sum_{j=1}^{k} d_H(\tilde{r}_{ij}, \tilde{M}_{j-})^2}, \quad i \in (1, 2, \cdots S) \tag{12}$$

where $D_i$ is the FO of the $i$th independent variable.

*Step 6:* Sort the candidate independent variables according to the $D_i$ values from smallest to largest. The independent variable which gives smallest FO value is regarded as the best solution.

### C. CASE STUDY

In order to demonstrate the computational process of the proposed FO, and exemplify its ability to provide a solution among sunken parts of the PF in nonconvex problems, an example is discussed in this section.

Consider the below six feasible solutions of a bi-objectives minimization problem: $a(0,1)$, $b(0.1,0.8)$, $c(0.2,0.9$, $d(0.56,0.56)$, $e(0.8,0.1)$ and $f(1,0)$, their distribution are shown in the objective space in Fig. 2. As shown in Fig. 2, this is a nonconvex problem, where $a$, $b$, $d$, $e$ and $f$ are all Pareto solutions, and $c$ is a dominated solution. $d$ is a solution which is located at the sunken part of the Pareto front. Here, we assume that two objectives are equally desired, thus, the preference vector is $\tilde{w} = [(0.5, 0.5, 0.5), (0.5, 0.5, 0.5)]$.

For the above problem, if the classic weighted sum method is utilized, then each solution is evaluated by the following equation,

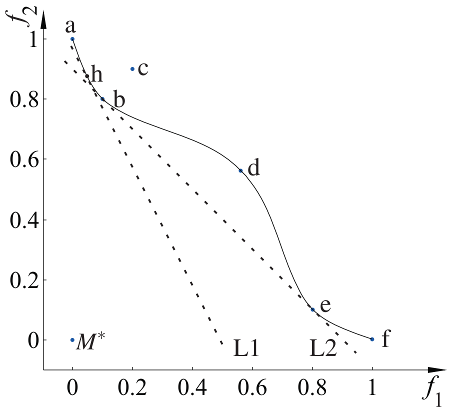


**FIGURE 2.** Diagram of the case study.

$$F = \sum_{i=1}^{k} w_i f_i \tag{13}$$

where, $k = 2$. If $w_1 = w_2 = 0.5$, then $F(b) = F(e) = 0.45$, solution $b$ and $e$ are selected as the best solutions. However, no matter which combination of weights is utilized, the solutions which is located in the sunken part of the Pareto front are not available. Fig. 2 describes this scenario. For a specific combination of the weighting factors, a contour line marked 'L1' or 'L2' in Fig. 2 results, the optimal solution with minimum $F$ will correspond to a Pareto solution which is the tangent point with the Pareto front, such as point $b$, $e$ and $h$. Unfortunately, no contour line will produce a tangent point with the sunken parts of the Pareto front between $b$ and $e$ [23]–[25].

If the proposed FO is utilized, the computational process is described as below.

*Step 1:* Normalize the objectives of the candidate solutions by equation (6), then the normalized weighted fuzzy decision matrix is described as,

$$\tilde{F} = \begin{bmatrix} (0,0,0) & (1,1,1) \\ (0.1,0.1,0.1) & (0.8,0.8,0.8) \\ (0.2,0.2,0.2) & (0.9,0.9,0.9) \\ (0.56,0.56,0.56) & (0.56,0.56,0.56) \\ (0.8,0.8,0.8) & (0.1,0.1,0.1) \\ (1,1,1) & (0,0,0) \end{bmatrix} \tag{14}$$

*Step 2:* The fuzzy utility matrix is obtained by applying equation (9),

$$\tilde{R} = \begin{bmatrix} (0,0,0) & (0.5,0.5,0.5) \\ (0.05,0.05,0.05) & (0.4,0.4,0.4) \\ (0.1,0.1,0.1) & (0.45,0.45,0.45) \\ (0.28,0.28,0.28) & (0.28,0.28,0.28) \\ (0.4,0.4,0.4) & (0.05,0.05,0.05) \\ (0.5,0.5,0.5) & (0,0,0) \end{bmatrix} \tag{15}$$

*Step 3:* Obtain the fuzzy ideal solution $\tilde{M}^*$ by equation (10) and (11),

$$\tilde{M}^* = [(0,0,0), (0,0,0)] \tag{16}$$

Here, the fuzzy ideal solution is marked by $M^*$ in the objective space, as shown in Fig. 2.

*Step 4:* Compute the Hamming distance between the $i$th solution and $\tilde{M}^*$ by equation (12),

$$D = [1; 0.8062; 0.922; 0.792; 0.8062]; \tag{17}$$

Thus, the solution $d$ which gives the smallest FO is selected as the best solution. The FO of a solution is essentially the Hamming distance between its fuzzy utility value and the fuzzy ideal solution, thus, its able to provide a solution among the sunken parts of the PF in nonconvex problem.

Consider solution $c$, which is dominated by solution $b$. Its FO is 0.922. For a minimization problem, if solution $\alpha$ is dominated by $\beta$, that means all objectives of $\beta$ are not bigger than those of $\alpha$. Thus, no matter which preference vector is utilized, the Hamming distance between the fuzzy utility value of $\alpha$ and the fuzzy ideal solution always larger than that of the $\beta$. That is, the solution which gives the minimum FO must be an Pareto solution.

## IV. FO BASED MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION

In this section, an interactive evolutionary approach is proposed for solving the MOPs. It inherit the basic structure of the PSO and evaluate the particles by the FO, thus we named it as FMOPSO.

### A. THE PARTICLE SWARM OPTIMIZATION ALGORITHM

PSO is a population stochastic algorithm which is originally designed for single objective optimization, It has been used to solve a range of optimization problems [26]–[29]. Wang *et al.* employed the PSO to search the optimum operating point of the auxiliary power unit under three kinds of constraints, where the PSO algorithm was programmed in Matlab language and executed on a rapid control prototyping toolkit [30]. Kim and Lee employed the PSO to optimize the trajectory of the manipulator, where the $D$-dimensional modification vector was encoded into a particle for PSO and the cost for each step was used for initializing the particles. The cost was calculated by multiplying the degree of a constraint violation with a weighting factor [31]. Furthermore, PSO was applied as a design tool for a parasitically coupled microstrip antenna array in [32], where PSO was employed to determine the shape of microstrip antennas with limited geometrical constraints allowing the optimization process to freely develop unique non intuitive designs for WLAN applications. The PSO Antenna software developed consists of two main components. The first is the PSO software that implements the PSO algorithms and the second is a Linking Program that connects the PSO software to either in-house or commercial antenna simulation software to allow the antenna performance and fitness function to be evaluated for a particular antenna geometry [32].

The PSO is initialized with a population of random solutions (named as particle swarm in PSO) and searches for optimum solutions by updating generations. In PSO, all particles fly through the problem space by following the optimum particles, each particle accelerates in the direction of its own personal best solution found so far, as well as in the direction of the global best position discovered so far by any of the particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, exploring the region more thoroughly in the process [33]–[35].

For a $d$ dimensions minimization problem, the swarm size is $N$, each individual particle ($1 \le i \le N$) has the following attributes: a current position $p_j^i(k)$ in the search space $\Omega^d$, where $j$ is the $j$th dimension of the particle and $j = 1, \cdots, d$; a current velocity $v_j^i(k)$; and a personal best position $Pb^i = [pb_1^i, \cdots, pb_d^i]$. During each iteration, each particle $i$ in the swarm is updated in the $j$th dimension using the velocity update formula (18) and the position update formula (19).

$$v_j^i(t+1) = \underbrace{\mu \cdot v_j^i(t)}_{velocity\ reference} + \underbrace{\tau_1 r_1 [pb_j^i(t) - p_j^i(t)]}_{individual\ optima\ reference} + \underbrace{\tau_2 r_2 [pg_j(t) - p_j^i(t)]}_{global\ optima\ reference} \tag{18}$$

$$p_j^i(t+1) = p_j^i(t) + v_j^i(t+1) \tag{19}$$

The velocity update formula can be divided into three subformula, the current velocity reference formula, the individual optima reference formula and the global optima formula, as shown in the equation (18), where $\mu$ is the inertia weight, $r_1, r_2$ are two random values in the range [0, 1], $\tau_1, \tau_2$ are the acceleration coefficients. $Pg = [pg_1, \cdots, pg_d]$ is the global optimal position.

The individual best position of each particle is updated by,

$$Pb^i(t+1) = \begin{cases} Pb^i(t), & f[P^i(t+1)] \ge f[Pb^i(t)] \\ P^i(t+1), & f[P^i(t+1)] < f[Pb^i(t)] \end{cases} \tag{20}$$

where, the global best position $Pg(t)$ is defined as,

$$Pg(t) = \arg\min f[Pb^i(t)] \quad 1 \le i \le N \tag{21}$$

After the positions of all the particles are evaluated, and corrections are made to the positions of $Pb^i$ and $Pg$, repetition of this cycle until the desired objective value is obtained or the maximum iterations number is reached.

### B. OVERALL COMPUTATIONAL FLOW OF THE FMOPSO

In order to accommodate the FO to multi-objective optimization problems, the FMOPSO - a variant of the PSO is proposed. It inherits the basic structure of the PSO and modified in the mechanism of individual optima and global optima reference update, where particles are evaluated by their FO values, rather than the fitness of the objective function. The overall computational flow of the proposed FMOPSO is presented by the pseudo code as shown in below, where the individual optima and global optima reference update operations are described in detailed (highlighted with bold italic fonts).

```
Algorithm FMOPSO($k, d, N, t_{max}, \mu, \tau_1, \tau_2$)
Parameter description:
  $k$,        number of the objective functions;
  $d$,        number of the independent variables;
  $N$,        swarm size;
  $t_{max}$,  maximum iteration number;
  $\mu$,      inertia weights, $\mu \in [0, 1]$;
  $\tau_1$,   acceleration coefficient 1;
  $\tau_2$,   acceleration coefficient 2.
Initialization: Initialize a $N*d$ matrix $P[N][d] \leftarrow rand \in \mathbb{R}$, each row $P_i$ of this matrix is a $d$ dimensions individual, iteration number $t \leftarrow 0$, velocity $v_j^i = 0$, history population of the $i$th individual particle $Ph_i = \varnothing$, where $i \in (1, \cdots, N), j \in (1, \cdots, d)$
while $t \leq t_{max}$ do
    Constraints handling: if $P_i$ is infeasible, then it is regenerated randomly.
    for $i = 1$ to $N$ do
        Individual optima reference update:
            Evaluate $P_i \cup Ph_i$ by FO:
            $\boldsymbol{D} = FO(P_i \cup Ph_i)$
            Update individual optima reference and $Ph_i$:
            $Pb_i = \min(\boldsymbol{D}_\upsilon), \upsilon = 1, 2, \cdots, t$, where $\upsilon$ denotes the size of the personal history population.
            $Ph_i = Ph_i \cup P_i$
    end for
    Global optima reference update:
        Evaluate $Pb_i, i = 1, 2, \cdots, N$ by FO:
        $\boldsymbol{D}_i = FO(Pb_i)$
        Update global optima reference:
        $Pg = \min(\boldsymbol{D}_i), i = 1, 2, \cdots, N$
    for $i = 1$ to $N$ do
        for $j = 1$ to $d$ do
            Velocity update:
            $v_j^i = \mu \cdot v_j^i + \tau_1 r_1 [pb_j^i - p_j^i] + \tau_2 r_2 [pg_j - p_j^i]$
            Position update:
            $p_j^i = p_j^i + v_j^i$
        end for
    end for
    $t$=$t$+1
end while
Output $P_g$ as the best solution
```

## V. NUMERICAL EXPERIMENTS AND RESULTS

### A. EXPERIMENTS ENVIRONMENTS AND TEST FUNCTIONS

The proposed FMOPSO is implemented in Java and executed on a computer equipped with a Intel E7400 2.8Ghz CPU and 4GB RAM. The operating system is Ubuntu Linux 13.10 64bit.

The Java JDK 1.5 development kit is employed. The developed FMOPSO software follows an object-oriented architecture, it consists of four classes, *algorithm*, *solution*, *function* and *operator*. Class *algorithm* represents the superclass of the proposed optimization engine FMOPSO, and it provides the execute entry of the software. As its name suggests, class *solution* represents a set of solution objects, it is a superclass aimed at describing the features of the solutions. All test functions considered in the numerical experiment are inherited from the class *function*, it provides two basic methods, evaluate the objective function and process the constraints. *Operator* is a superclass aimed at representing the basic operator of the FMOPSO, such as swarm initialization, velocity update, position update, et al.

To validate the performance of the FMOPSO, the results of FMOPSO are compared against two highly competitive PO based algorithms: NSGA-II [36] and MOPSO [10]. For fair comparison, the FMOPSO parameters used in the experiments are set as: $\mu = 0.4$, $\tau_1 = 2.0$, $\tau_2 = 2.0$, $t_{max}$ and $N$ are determined by the specific test function. As regards the parameters of the NSGA-II and MOPSO, we follow the settings of the original paper [10], [36].

Because of the results of the PO based algorithm is a set of Pareto optimal solutions while the results of the FMOPSO is a single solution, it is difficult to compare the results directly. In order to compare the results between the PO based approaches and FMOPSO objectively, the results of the PO based approaches are evaluate by a simple weighting based decision maker and thus a sole best compromise solution is derived [23]. Furthermore, due to the objective functions are in different scales, they cannot be directly applied the weighting based decision maker and should be converted to a similar non-dimensional scale firstly. Thus, each of the $k$ objective values is normalized to a value between 0 and 1 by the following equation,

$$\bar{f}_{ij}(\boldsymbol{x}) = \frac{f_{ij}(\boldsymbol{x}) - f_j^m}{f_j^M - f_j^m}, \quad j = 1, 2, \cdots, k \tag{22}$$

where $f_j^m = \min_{i=1,2,\cdots,S} f_{ij}(\boldsymbol{x}), f_j^M = \max_{i=1,2,\cdots,S} f_{ij}(\boldsymbol{x})$.

Six representative test functions ZDT1, ZDT3, CTP2, TNAK, VET3 and DTLZ7 are used to evaluate the performance of the tested algorithms. These test functions have different features in solution space and objective space, thus they are adequate for demonstrating the performance of the multi-objective optimization approaches.

For each algorithm, the experiments are performed with three different preference vectors $\tilde{w}_1, \tilde{w}_2$ and $\tilde{w}_3$, where for the bi-objective test functions (ZDT1, ZDT3, CTP2 and TNAK), the preference vectors are set as $\tilde{w}_1 = [(0.2, 0.2, 0.2), (0.8, 0.8, 0.8)]$, $\tilde{w}_2 = [(0.5, 0.5, 0.5), (0.5, 0.5, 0.5)]$ and $\tilde{w}_3 = [(0.8, 0.8, 0.8), (0.2, 0.2, 0.2)]$, while for the three-objective test functions (VET3 and DTLZ7), the preference vectors are set as $\tilde{w}_1 = [(0.2, 0.2, 0.2), (0.2, 0.2, 0.2), (0.6, 0.6, 0.6)]$, $\tilde{w}_2 = [(0.2, 0.2, 0.2), (0.6, 0.6, 0.6), (0.2, 0.2, 0.2)]$ and $\tilde{w}_3 = [(0.6, 0.6, 0.6), (0.2, 0.2, 0.2), (0.2, 0.2, 0.2)$. For all of the three approaches, the population size is set to $N = 500$ for bi-objective test functions, and $N = 800$ for three-objective test functions. The iterations number is set to $t_{max} = 100$.

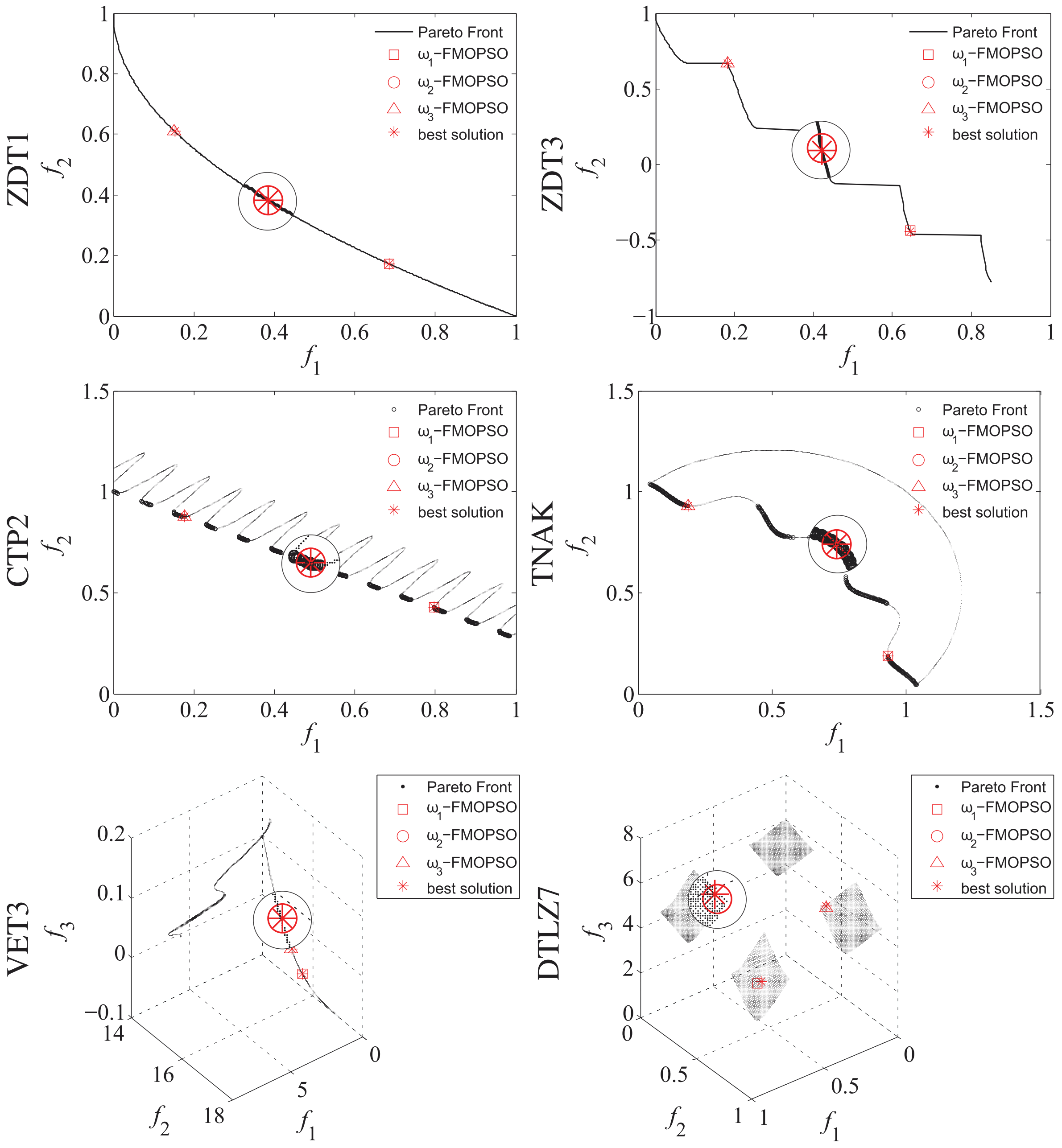


**FIGURE 3.** Experimental results of the proposed FMOPSO.

For the MOPSO, the size of the external repository is same as the population size.

### B. RESULTS ANALYSIS

The experimental result are analyzed on three aspects: quality of the solutions, robustness of the results and computational complexity.

#### 1) QUALITY OF SOLUTIONS

The experimental results of the FMOPSO, NSGA-II and MOPSO are graphically illustrated in Fig. 3, Fig. 4 and Fig. 5, respectively. In these figures, the *best solution* means the best compromise solution of the real Pareto front for the specific weight factors. Furthermore, to show the results more clearly, we use 2.5-times magnifying glasses to show the details of the results of $\tilde{w}_2$ for each test function and each optimization approach in Fig. 3, Fig.4 and Fig. 5.

As shown in Fig. 3, for most test functions, the results of the FMOPSO are coincide with the *best solutions*. For the DTLZ7, which is a three-objective function, the deviation between the results of the FMOPSO and the *best solutions* is slightly greater compared with that of the bi-objective functions. Furthermore, the results of the DTLZ7 shown in Fig. 4 and Fig. 5 reveal that the NSGA-II and MOPSO

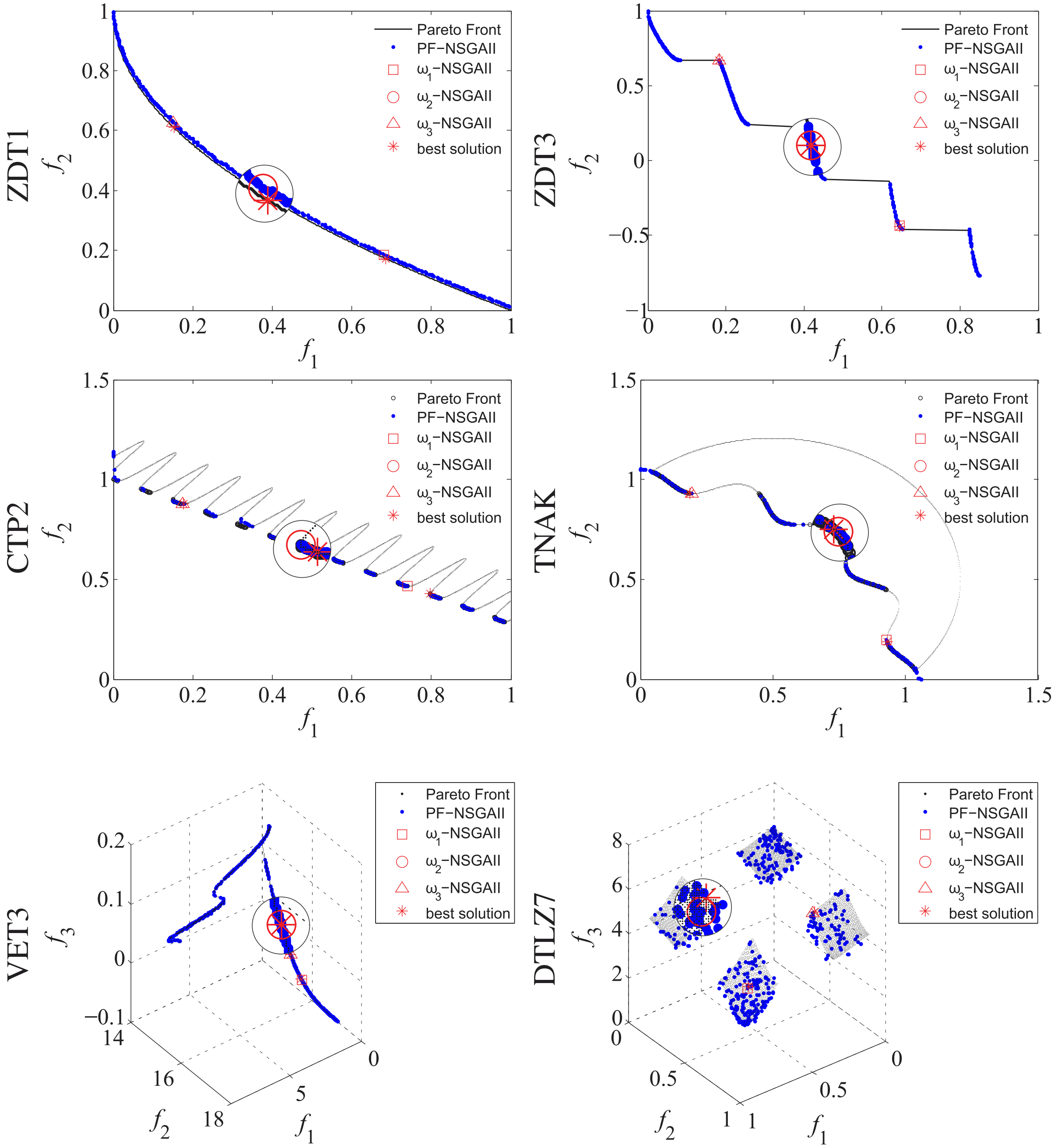


**FIGURE 4.** Experimental results of the NSGA-II.

perform more poorly on the DTLZ7. Fig. 4 reveals that the NSGA-II performs poorly on ZDT1, CTP2, TNAK and DTLZ7, although it obtains fairly good results on ZDT3 and VET3. As for the MOPSO, the results shown in Fig. 5 reveals that it obtains comparative results compared with the NSGA-II for all test functions except for CTP2, which is quite poor.

In order to quantitatively analyze the results of the three approaches, for each preference vector, the distance between the result and the *best solution* in objective space is calculated according to the following equation,

$$d = \sqrt{\sum_{i=1}^{k} [f_i(\boldsymbol{x}) - f_i(\boldsymbol{x}_{best})]^2} \tag{23}$$

where $k$ is the dimension of the objective space. The quantitative results of the three approaches for each preference vector are shown in Table 1, where the gray colored background is used to point out the algorithms which obtaining the best values.

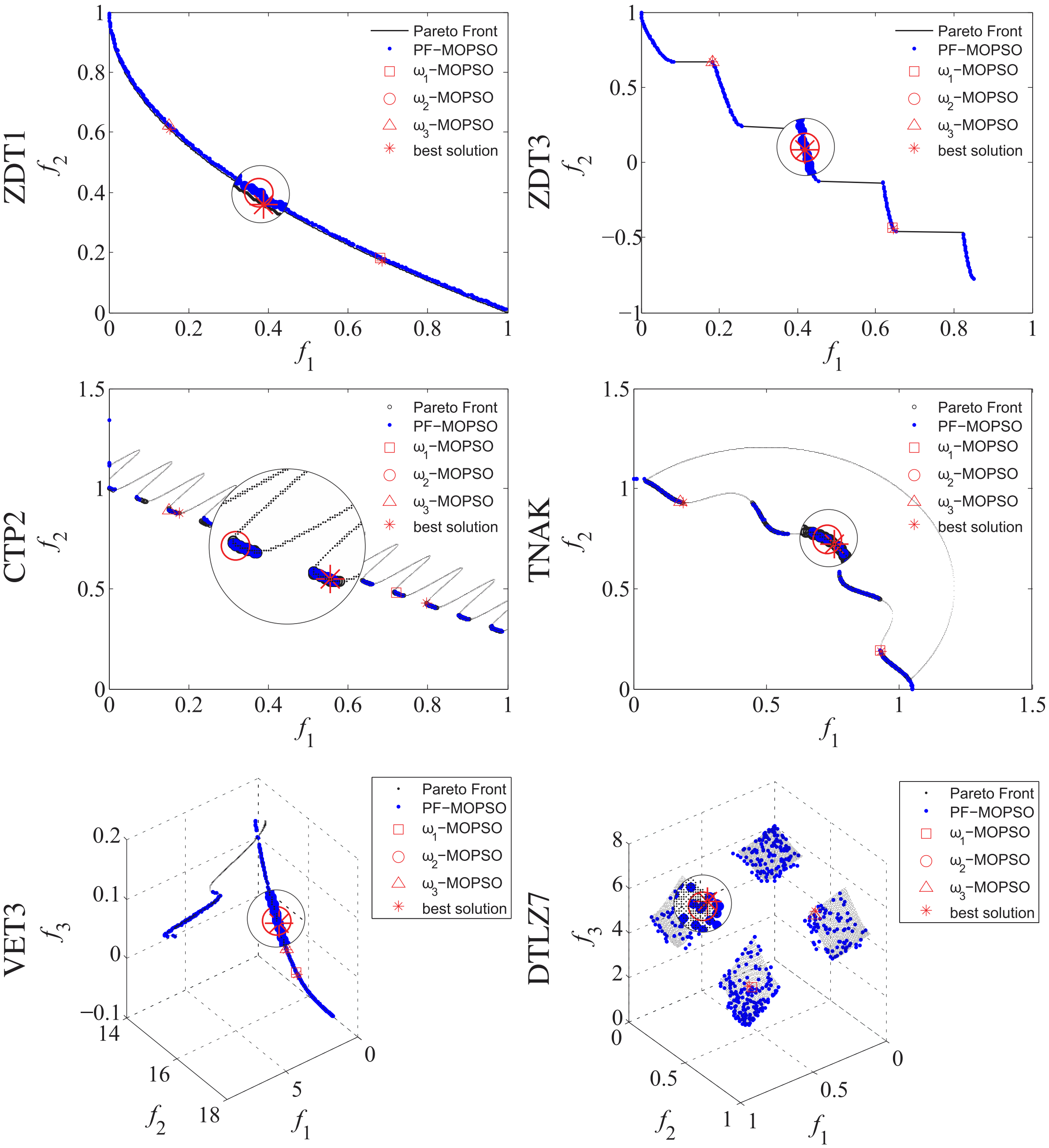


**FIGURE 5.** Experimental results of the MOPSO.

Table 1 reveals that for the $\tilde{w}_1$, the proposed FMOPSO obtains best values in 4 out of 6 test functions, the other two best values are obtained by the MOPSO algorithm. For the $\tilde{w}_2$, the FMOPSO obtains best values in 5 out of 6 test functions, the only exception is the ZDT3 function, its best value is obtained by the NSGA-II algorithm. For the $\tilde{w}_2$, the FMOPSO obtains best values for all of the 6 test function.

Based on the above analysis, we can reach a conclusion that the proposed FMOPSO is superior to the NSGA-II and MOPSO on aspect of quality of solutions.

### 2) ROBUSTNESS OF THE RESULTS

As a population based heuristic optimization approach, there may exist randomness in the result of the FMOPSO. In order to evaluate the robustness of the FMOPSO, all of the three approaches are run 20 times on each test function, where for the bi-objective test functions, the preference vector is set as $\tilde{w} = [(0.5, 0.5, 0.5), (0.5, 0.5, 0.5)]$, while for the three-objective test functions, the preference vector is set as $\tilde{w} = [(0.2, 0.2, 0.2), (0.6, 0.6, 0.6), (0.2, 0.2, 0.2)]$. The results of each test function are presented in box-and-whisker plot, as shown in Fig. 6. Here, the box-and-whisker

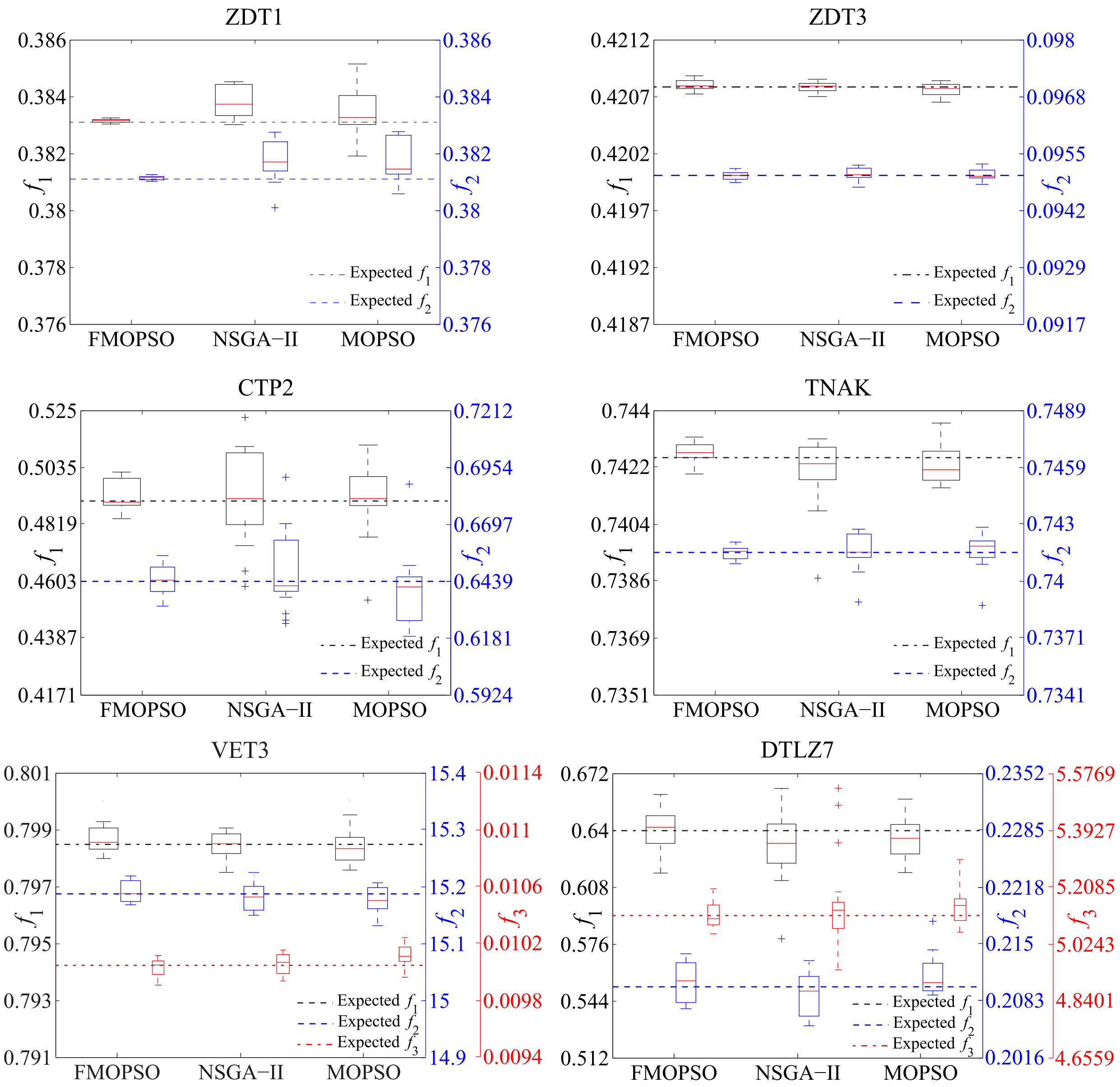


**FIGURE 6.** Box-and-whisker plots for the results of 20 times experiments.

**TABLE 1.** Quantitative Results of the Three Approaches.

| Test function | $\tilde{w}_1$ | | | $\tilde{w}_2$ | | | $\tilde{w}_3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | FMOPSO | NSGA-II | MOPSO | FMOPSO | NSGA-II | MOPSO | FMOPSO | NSGA-II | MOPSO |
| ZDT1 | 0.0005 | 0.0129 | 0.0118 | 0.0004 | 0.0163 | 0.0158 | 0.0017 | 0.0154 | 0.0138 |
| ZDT3 | 0.0039 | 0.0052 | 0.0086 | 0.0082 | 0.0004 | 0.0050 | 0.0007 | 0.0024 | 0.0015 |
| CTP2 | 0.0000 | 0.0691 | 0.0940 | 0.0013 | 0.0225 | 0.1166 | 0.0000 | 0.0022 | 0.0304 |
| TNAK | 0.0035 | 0.0100 | 0.0028 | 0.0004 | 0.0095 | 0.0123 | 0.0030 | 0.0039 | 0.0130 |
| VET3 | 0.0024 | 0.0127 | 0.0329 | 0.0000 | 0.0008 | 0.0139 | 0.0034 | 0.0039 | 0.0120 |
| DTLZ7 | 0.0503 | 0.0463 | 0.0182 | 0.0263 | 0.1626 | 0.0692 | 0.0422 | 0.0179 | 0.0879 |

plot shows the statistical characteristics of the experimental results clearly, it summarizes data using the median, upper and lower quartiles and the extreme (minimum and maximum) values, it allows you to see important characteristics of the data at a glance.

In Fig. 6, the statistical characteristics of the objective functions for 6 test functions over 20 times independent runs are shown. As can be seen in Fig. 6, the FMOPSO obtains better median, minimum, maximum, lower quartile and upper quartile values than those of the other approaches for ZDT1,

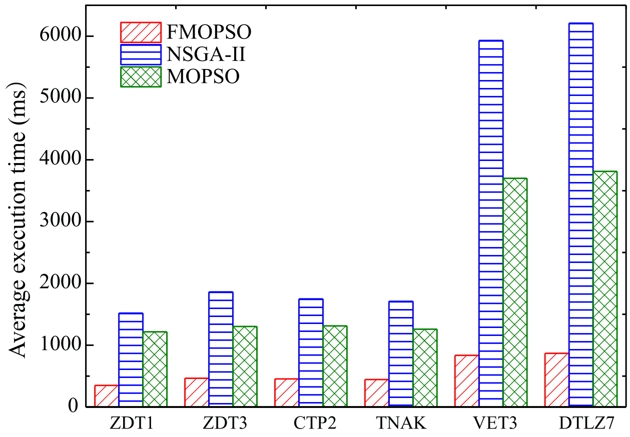


**FIGURE 7.** Average execution time of three approaches.

CTP2 and TNAK. For the ZDT3, VET3 and DTLZ7 test functions, three approaches obtain competitive results. Furthermore, for the results of the FMOPSO, there are no outlier for all the objective functions of the 6 test functions. While for the NSGA-II and MOPSO, there are many outliers, such as the $f_2$ of ZDT1 for NSGA-II, the two objective functions of CTP2 for both algorithms.

Based on the above analysis, we can reach a conclusion that the proposed FMOPSO shows better robustness than the NSGA-II and MOPSO.

#### 3) COMPUTATIONAL COMPLEXITY

The average execution time of the three approaches on six test functions are also analyzed statistically. As shown in Fig. 7, NSGA-II is the approach which has the longest average execution time for all six test functions. The MOPSO has slightly shorter execution times than that of NSGA-II. However, the average execution time of the FMOPSO is only about a quarter of that of NSGA-II for the bi-objective test functions and one in seven for the three-objective test functions. This is because that the time consuming non-dominated sorting process is avoided in FMOPSO and the time complexity of this process is increased sharply with the increase of numbers of the objective functions [36]. As regards the MOPSO, it adopts an external elitist archive to retain the non-dominated solutions and introduces the adaptive grid procedure to maintain the diversity of the population, the time complexity of these operation are slightly lower than that of the NSGA-II [10]. Thus, its average execution time is slightly shorter than that of the NSGA-II.

Obviously, since the time consuming non-dominated sorting process is avoided, the computational complexity of the FMOPSO is reduced significantly.

## VI. PRACTICAL CASE STUDY

To further validate the efficacy and practicality of the proposed FMOPSO, a case study based on a real-word problem is performed.

### A. APU OPERATING POINT MOP

Auxiliary power units (APUs) are widely used for electric power generation in various types of electric vehicles, improvements in fuel economy and emissions of these vehicles directly depend on the operating point of the APUs. To balance the conflicting goals of fuel consumption and emissions reduction, the APU operating point optimization problem is formulated as a constrained multi-objective optimization problem which contains two independent variables $(n, T)$ and four competing objectives FE cost, HC emissions, CO emissions and NOx emissions [37],

$$\begin{aligned} \mathit{minimize}: F(n,T) = [&f_{FE}(n,T), f_{HC}(n,T),\\ &f_{CO}(n,T), f_{NOx}(n,T)] \end{aligned} \tag{24}$$

$$\mathit{subject\ to}: \begin{cases} \max(n_{\min}, \Xi_x(f_{tm}(n), g(P_{low}, n))) \le n \le \\ \qquad \min(n_{\max}, \Xi_x(T = T_{\min}, g(P_{high}, n))) \\ \max(T_{\min}, g(P_{low}, n)) \le T \le \min(f_{tm}(n), \\ \qquad\qquad g(P_{high}, n)) \end{cases} \tag{25}$$

For the APU operating point multi-objective optimization problem, based on the same experimental facility and procedure as in [37], an experiment-based case study is performed.

### B. EXPERIMENTAL RESULT ANALYSIS

In the experiments, HWFET, NEDC and JPN1015 driving cycle are utilized to evaluate the optimization results of the FMOPSO. The fuel consumption(FC), HC, CO and NOx emissions of the FMOPSO for the three driving cycles are present in Table 2, where the experimental results of the FMOPSO are compared with that of the AMODE reported in [37]. Since the weights of the FC, HC, CO and NOx were set as 0.4, 0.2, 0.1 and 0.3 in AMODE, the preference vectors of the FMOPSO are set as $\tilde{w} = [(0.4, 0.4, 0.4), (0.2, 0.2, 0.2), (0.1, 0.1, 0.1), (0.3, 0.3, 0.3)]$. In Table 2, the third column of each optimization objective represents the comparison results (CR), CR= $R_{AMODE}$ - $R_{FMOPSO}$, where $R_{AMODE}$ and $R_{FMOPSO}$ are the experimental results of the two approaches, respectively.

Since the APU operating point multi-objective optimization problem is a minimization problem and the smaller FC, HC, CO and NOx values are desired, the positive CR means the FMOPSO obtains the better result. As shown in Table 2, compared with the AMODE, the proposed FMOPSO obtains better results on majority driving cycles and optimization objectives, such as the FC for all driving cycles, the CO for HWFET and JPN1015, NOx for NEDC and JPN1015. As for the average results off all driving cycles, the FMOPSO obtains better or equal results on aspects of FC, CO and NOx emissions, which indicates that the FMOPSO is capable to solve the real-word multi-objective optimization problem and shows better performance on aspects of quality of solutions and robustness.

**TABLE 2.** Experimental Results of the AMODE and FMOPSO.

| Cycles | FC(L/100km) | | | HC(g/km) | | | CO(g/km) | | | NOx(g/km) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FMOPSO | AMODE | CR | FMOPSO | AMODE | CR | FMOPSO | AMODE | CR | FMOPSO | AMODE | CR |
| HWFET | 5.165 | 5.167 | 0.002 | 0.067 | 0.067 | 0 | 0.926 | 0.934 | 0.008 | 0.074 | 0.061 | -0.013 |
| NEDC | 6.646 | 6.680 | 0.034 | 0.114 | 0.114 | 0 | 1.011 | 1.009 | -0.002 | 0.085 | 0.086 | 0.001 |
| JPN1015 | 6.511 | 6.602 | 0.091 | 0.125 | 0.117 | -0.008 | 1.001 | 1.096 | 0.095 | 0.083 | 0.097 | 0.014 |
| Average | 6.107 | 6.150 | 0.043 | 0.102 | 0.099 | -0.003 | 0.979 | 1.013 | 0.034 | 0.081 | 0.081 | 0 |

## VII. CONCLUSION

In order to overcome the limitations of the PO in solving multi-objective optimization problems, a new definition of optimality: FO is presented, and further, the FMOPSO is proposed. The innovation of this paper can be summarized as follows:

1. The number of improved objectives and the extent of the improvements are both considered in the FO, the linguistic preference information of objectives are represented by the fuzzy triangular numbers and processed by fuzzy mathematics. Furthermore, the proposed FO is able to provide a solution among sunken parts of the PF in nonconvex problems;
2. Through introduce the FO definition into the traditional PSO, a heuristic multi-objective optimization approach FMOPSO is proposed.

To validate the performance and practicality of the FMOPSO, experiments were carried out on 6 representative test functions and APU operating point MOP. The experimental results show that the proposed FMOPSO is superior to the existing approaches on aspects of quality of solutions, robustness and computational complexity.

## APPENDIX A

### DEFINITION 1

As an example of the fuzzy subset and membership function, let fuzzy subset $\tilde{A}$ denotes the “young people" and its member ship function is,

$$\mu_{\tilde{A}}(x)=\begin{cases}1 & 15\leq x\leq 25\\ \frac{1}{1+(\frac{x-25}{5})^2} & 25<x\leq 35\end{cases}\tag{26}$$

Thus, the degree that a 30 years old people belongs to the “young people" is $\mu_{\tilde{A}}(30)=0.5$.

### DEFINITION 2

As an example of the triangular fuzzy number, Fig. 8 shows the diagram of $\tilde{M}$=(1, 1.5, 3), and the piecewise linear membership function of $\tilde{M}$ is,

$$\mu_{\tilde{M}}(x)=\begin{cases}L(x)=2x-2 & 1\leq x\leq 1.5\\ R(x)=-\dfrac{2}{3}x+2 & 1.5<x\leq 3\end{cases}\tag{27}$$

### DEFINITION 3

The black thich lines in Fig. 9 shows the diagram of the minimum fuzzy subset between $\tilde{M}$=(1, 1.5, 3) and $\tilde{N}$=(0.5, 2, 2.5).

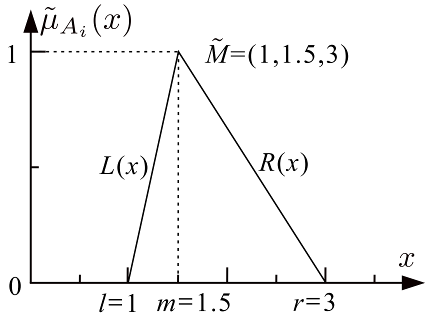


**FIGURE 8.** A triangular fuzzy number.

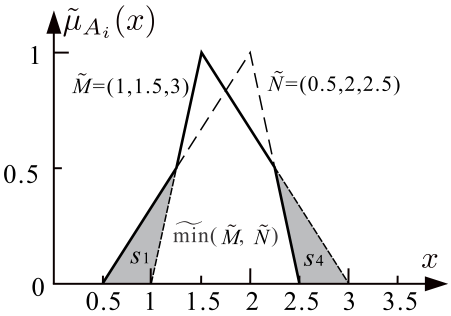


**FIGURE 9.** Diagram of the minimum fuzzy subset and Hamming distance.

The member function of $\widetilde{min}(\tilde{M},\tilde{N})$ is,

$$\mu_{\widetilde{min}(\tilde{M},\tilde{N})}(x)=\begin{cases}\dfrac{2}{3}x-\dfrac{1}{3} & 1\leq x\leq 1.25\\ 2x-2 & 1.25<x\leq 1.5\\ -\dfrac{2}{3}x+2 & 1.5<x\leq 2.25\\ -2x+5 & 2.25<x\leq 2.5\end{cases}\tag{28}$$

### DEFINITION 4

As an example of the Hamming distance between two fuzzy subset, Fig. 9 shows the diagram of the Hamming distance between $\tilde{M}$ and $\widetilde{min}(\tilde{M},\tilde{N})$, that is,

$$\begin{aligned}d_H(\tilde{M},\widetilde{min}(\tilde{M},\tilde{N}))&=\int_{x\in R}|\mu_{\tilde{M}}(x)-\mu_{\widetilde{min}(\tilde{M},\tilde{N})}(x)|dx\\&=s_1+s_4=0.25\end{aligned}\tag{29}$$

where $s_1$ and $s_4$ are the area of the shaded areas which constitute the geometric metric of Hamming distance.

### DEFINITION 5

Here, an example about the calculation of the normalized fuzzy triangular number is given, consider the following two triangular fuzzy number $\tilde{f}_{11} = (1, 1.5, 3)$ and $\tilde{f}_{21} = (0.5, 2, 2.5)$, then $r_1^{\max} = 3$, $m_1^{\max} = 2$ and $l_1^{\max} = 1$. According to equation (6), the normalized fuzzy triangular number of $\tilde{f}_{11}$ is,

$$\begin{aligned}\tilde{f}_{\Delta 11} &= (\frac{l_{11}}{r_1^{\max}}, \frac{m_{11}}{m_1^{\max}}, \frac{r_{11}}{l_1^{\max}} \wedge 1)\\ &= (\frac{1}{3}, \frac{3}{4}, 1)\end{aligned} \tag{30}$$

Similarly, the normalized fuzzy triangular number of $\tilde{f}_{21}$ is,

$$\begin{aligned}\tilde{f}_{\Delta 21} &= (\frac{l_{21}}{r_1^{\max}}, \frac{m_{21}}{m_1^{\max}}, \frac{r_{21}}{l_1^{\max}} \wedge 1)\\ &= (\frac{1}{6}, 1, 1)\end{aligned} \tag{31}$$

## REFERENCES

[1] M. G. Villarreal-Cervantes, A. Rodríguez-Molina, C.-V. García-Mendoza, O. Peñaloza-Mejía, and G. Sepúlveda-Cervantes, "Multi-objective on-line optimization approach for the DC motor controller tuning using differential evolution," *IEEE Access*, vol. 5, pp. 20393–20407, 2017.

[2] X. Zhou, J. Zhou, C. Yang, and W. Gui, "Set-point tracking and multi-objective optimization-based PID control for the goethite process," *IEEE Access*, vol. 6, pp. 36683–36698, 2018.

[3] D. Wang and K. Lu, "Design optimization of hydraulic energy storage and conversion system for wave energy converters," *Protection Control Mod. Power Syst.*, vol. 3, no. 1, pp. 1–9, 2018.

[4] A. I. Aravanis, M. R. S. Bhavani, P. D. Arapoglou, G. Danoy, P. G. Cottis, and B. Ottersten, "Power allocation in multibeam satellite systems: A two-stage multi-objective optimization," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3171–3182, Jun. 2015.

[5] M. Khari, P. Kumar, D. Burgos, and R. G. Crespo, "Optimized test suites for automated testing using different optimization techniques," *Soft Comput.*, vol. 22, no. 24, pp. 8341–8352, 2018.

[6] S. Arora and S. Singh, "An effective hybrid butterfly optimization algorithm with artificial bee colony for numerical optimization," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 4, pp. 14–21, 2017.

[7] J. Meza, H. Espitia, C. Montenegro, and E. Giménez, and R. González-Crespo, "MOVPSO: Vortex multi-objective particle swarm optimization," *Appl. Soft Comput.*, vol. 52, pp. 1042–1057, Mar. 2017.

[8] R. E. Haber, G. Beruvides, R. Quiza, and A. Hernandez, "A simple multi-objective optimization based on the cross-entropy method," *IEEE Access*, vol. 5, pp. 22272–22281, 2017.

[9] D. Zhou, J. Wang, B. Jiang, H. Guo, and Y. Li, "Multi-task multi-view learning based on cooperative multi-objective optimization," *IEEE Access*, vol. 6, pp. 19465–19477, 2018.

[10] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[11] A. Sadollah, H. Eskandar, and J. H. Kim, "Water cycle algorithm for solving constrained multi-objective optimization problems," *Appl. Soft Comput.*, vol. 27, pp. 279–298, Feb. 2015.

[12] Y. Qi, Z. Hou, M. Yin, H. Sun, and J. Huang, "An immune multi-objective optimization algorithm with differential evolution inspired recombination," *Appl. Soft Comput.*, vol. 29, pp. 395–410, Apr. 2015.

[13] Z. Liang, R. Song, Q. Lin, Z. Du, J. Chen, Z. Ming, and J. Yu, "A double-module immune algorithm for multi-objective optimization problems," *Appl. Soft Comput.*, vol. 35, pp. 161–174, Oct. 2015.

[14] A. Ahmadi, "Memory-based adaptive partitioning (MAP) of search space for the enhancement of convergence in Pareto-based multi-objective evolutionary algorithms," *Appl. Soft Comput.*, vol. 41, pp. 400–417, Apr. 2016.

[15] X. Dai, X. Yuan, and Z. Zhang, "A self-adaptive multi-objective harmony search algorithm based on harmony memory variance," *Appl. Soft Comput.*, vol. 35, pp. 541–557, Oct. 2015.

[16] P. Bagga, A. Joshi, and R. Hans, "QoS based Web service selection and multi-criteria decision making methods," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 5, no. 4, pp. 113–121, 2019.

[17] C. Zhu, L. Xu, and E. D. Goodman, "Generalization of pareto-optimality for many-objective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 299–315, Apr. 2016.

[18] C. He, L. Li, Y. Tian, X. Zhang, R. Cheng, Y. Jin, and X. Yao, "Accelerating large-scale multi-objective optimization via problem reformulation," *IEEE Trans. Evol. Comput.*, to be published. doi: 10.1109/TEVC.2019.2896002.

[19] A. K. Das and D. K. Pratihar, "A novel approach for neuro-fuzzy system-based multi-objective optimization to capture inherent fuzziness in engineering processes," *Knowl.-Based Syst.*, vol. 175, pp. 1–11, Jul. 2019.

[20] D. Alanis, P. Botsinis, Z. Babar, H. V. Nguyen, D. Chandra, S. X. Ng, and L. Hanzo, "Quantum-aided multi-objective routing optimization using back-tracing-aided dynamic programming," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7856–7860, Aug. 2018.

[21] M. Farina and P. Amato, "A fuzzy definition of 'optimality' for many-criteria optimization problems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Hum.*, vol. 34, no. 3, pp. 315–326, May 2004.

[22] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, and B. De Baets, "A historical account of types of fuzzy sets and their relationships," *IEEE Trans. Fuzzy Syst.*, vol. 24, no. 1, pp. 179–194, Feb. 2016.

[23] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley, 2001.

[24] H. Seada and K. Deb, "Non-dominated sorting based multi/many-objective optimization: Two decades of research and application," in *Multi-Objective Optimization*. Singapore: Springer, 2018, pp. 1–24.

[25] X. Li, M. G. Epitropakis, K. Deb, and A. Engelbrecht, "Seeking multiple solutions: An updated survey on niching methods and their applications," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 518–538, Aug. 2017.

[26] J. H. Lee, J.-Y. Song, D.-W. Kim, J.-W. Kim, Y.-J. Kim, and S.-Y. Jung, "Particle swarm optimization algorithm with intelligent particle number control for optimal design of electric machines," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1791–1798, Feb. 2018.

[27] H. Han, W. Lu, L. Zhang, and J. Qiao, "Adaptive gradient multiobjective particle swarm optimization," *IEEE Trans. Cybern.*, vol. 48, no. 11, pp. 3067–3079, Nov. 2018.

[28] X. Wang, G. Wang, and Y. Wu, "An adaptive particle swarm optimization for underwater target tracking in forward looking sonar image sequences," *IEEE Access*, vol. 6, pp. 46833–46843, 2018.

[29] H. Zhou, X. Wang, and N. Cui, "A novel reentry trajectory generation method using improved particle swarm optimization," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3212–3223, Apr. 2019.

[30] Y. Wang, Y. Shen, X. Yuan, and Y. Yang, "Operating point optimization of auxiliary power unit based on dynamic combined cost map and particle swarm optimization," *IEEE Trans. Power Electron.*, vol. 30, no. 12, pp. 7038–7050, Dec. 2015.

[31] J.-J. Kim and J.-J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 620–631, Jun. 2015.

[32] A. A. Minasian and T. S. Bird, "Particle swarm optimization of microstrip antennas for wireless communication systems," *IEEE Trans. Antennas Propag.*, vol. 61, no. 12, pp. 6214–6217, Dec. 2013.

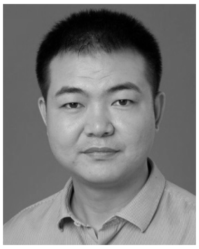
[33] C. Lagos, G. Guerrero, E. Cabrera, A. Moltedo-Perfetti, F. Johnson, and F. Paredes, "An improved particle swarm optimization algorithm for the VRP with simultaneous pickup and delivery and time windows," *IEEE Latin Amer. Trans.*, vol. 16, no. 6, pp. 1732–1740, Jun. 2018.

[34] X. Yu, W.-N. Chen, T. Gu, H. Zhang, H. Yuan, S. Kwong, and J. Zhang, "Set-based discrete particle swarm optimization based on decomposition for permutation-based multiobjective combinatorial optimization problems," *IEEE Trans. Cybern.*, vol. 48, no. 7, pp. 2139–2153, Jul. 2018.

[35] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K.-C. Wong, and J. Zhang, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 32–46, Feb. 2018.

[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[37] Y. Shen and Y. Wang, "Operating point optimization of auxiliary power unit using adaptive multi-objective differential evolution algorithm," *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 115–124, Jan. 2017.



**YONGPENG SHEN** was born in Henan, China, in 1985. He received the M.S. degree in electrical engineering from the Zhengzhou University of Light Industry, Zhengzhou, China, in 2010, and the Ph.D. degree in control science and engineering from Hunan University, Changsha, China, in 2015.

Since 2015, he has been with the Henan Key Lab of Information-Based Electrical Appliances, Zhengzhou University of Light Industry. His research interest includes optimization method and its application in electric and hybrid electric vehicle.

**GAORUI GE** was born in Henan, China, in 1992. He received the B.S. degree from the Henan Institute of Science and Technology, Xinxiang, China, in 2017. He is currently pursuing the M.S. degree with the School of Electrical Engineering, Zhengzhou University of Light Industry, Zhengzhou.

His current research interests include intelligent computation and its applications in power converter, and battery management system design.