New Method for Solving Full Fuzzy Quadratic Programming Problem

Xianfeng Ding, Jiaxin Li, Tingting Wei, Yiyuan Wang

***Abstract*—The problem of programming with fuzzy coefficients and fuzzy variables is more general. This paper studies the quadratic programming problem whose parameters are all fuzzy numbers based on the A-PSO algorithm. Especially, the four operations of triangular fuzzy numbers are extended and an improved sorting method for triangular fuzzy numbers is proposed that takes into consideration the diverse membership of each point. Moreover, the exact solution steps of the fully fuzzy quadratic programming problem are given by the new method proposed in this paper. Numerical examples are submitted to compare and analyze the algorithm and results, which demonstrate the efficiency and effectiveness of the proposed method. Finally, the method is increased to solve all fuzzy nonlinear programming problems.**

***Index Terms*—full fuzzy quadratic programming problem, triangular fuzzy numbers, A-PSO algorithm**

I. Introduction

IN the real world, different mathematical models can be constructed according to separate research objects of the optimization problems. The classification of optimization models can be determined by enabling the target to evaluate definite criteria or multiple criteria of criteria and whether there are limitation conditions. When the parameters in the model are accurate data, it is regarded as a deterministic model [1]. When it involves random parameters, it regards as an uncertain model.

In the deterministic optimization problem, the parameters and constraints involved in the model are fixed. Additionally, the decision maker's choice is single. However, there is frequently uncertain information in life, such as incorrect data and ambiguous language. Thus, the uncertainty model can be closer to the application of actual problems. Establishing mathematical models with fuzzy numbers is a method of selection for uncertain programming. In 1970, Bellman and Zadeh proposed the decision model in fuzzy environment [2], which supplied the basis for the study of fuzzy optimization theory and method. Since then, the study of fuzzy programming has become a popular field. Many scholars have conducted relevant studies on optimization models, such as linear programming problems [3-7], multi-objective programming problem [15] and nonlinear programming problem [12-26] in a fuzzy environment.

The fuzzy linear programming problem (FLP problem) is mature in theory and application. In 2013, Kaur et al. proposed an unrestricted operational rule for LR flat fuzzy numbers [3]. They made the operation of fuzzy numbers universal and were implemented to solve problems in fully fuzzy linear programming. In 2020, Ranjbar et al. introduced the hesitation fuzzy number into the fuzzy linear programming problem [4]. In 2020, Boris et al. expanded the dictionary method to the solution of fuzzy linear programming problem [5]. In 2020, Kumar et al. proposed an effective method based on simple linear fractional programming to solve the problems of fuzzy linear fractional programming in daily life [6]. In 2021, Stanojevi et al. studied the fully fuzzy linear programming and proposed a membership function model that could numerically describe the feasible target value fuzzy sets [7]. In different types of FLP problems, the coefficients or variables of the planning problems are generally set as fuzzy numbers. So, the total ambiguity problem has a more general meaning.

Quadratic programming problems involve forward-looking programming. It is typically applied in portfolio, production scheduling, engineering design and inventory management. Meanwhile, quadratic programming is a special nonlinear programming problem. It is essential to construct a bridge between linear programming and nonlinear programming problems. Because of the existence of uncertain information, fuzzy quadratic programming is proposed to solve this kind of problem. Fuzzy quadratic programming can also be exerted on the corresponding fields of quadratic programming, as can be observed in literature [8-11]. Research into fuzzy quadratic programming is still under development. According to the fuzziness of discrete parameters, it can be structured in different programming models. Variable parameters with fuzziness: In 2017, Mirmohseni et al. proposed a novel method to derive fuzzy quadratic programming problems with constraint coefficients and triangular fuzzy numbers on the right [12]. In 2019, Ghanbari et al. used the ABS algorithm to provide a general compromise solution for LR fuzzy linear systems. They extended it to the quadratic programming problem with fuzzy LR variables [13].

Coefficient parameters have fuzziness: In 2019, Ghanbari

Manuscript received May 11, 2022; revised February 27, 2023. This work was supported in part by the International Cooperation program of Chengdu City 2020-GH02-00023-H2.

Xianfeng Ding is an associate professor in the School of Science of Southwestern Petroleum University, Chengdu, 611756, Sichuan, China. (e-mail: fxxd@163.com).

Jiaxin Li, the corresponding author, is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu, 611756, Sichuan, China. (e-mail: lljjxx_09@163.com).

Tingting Wei is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu, 611756, Sichuan, China. (e-mail: 1558862556@qq.com).

Yiyuan Wang is a postgraduate student in the School of Science, Southwest Petroleum University, Chengdu, 611756, Sichuan, China. (e-mail: 1743821680@qq.com).

applied the improved Kerre's method to solve the quadratic programming problem of triangular fuzzy numbers with coefficients [14]. In 2020, Niswatus et al. studied the definition and algorithm of the triangular fuzzy number and proposed a new completion method [15]. They turned lone fuzzy quadratic programming into simple quadratic multi-objective programming.

Problems with fuzziness on constraints: In 2011, Cruz proposed two methods for resolving linear problems with uncertain constraint sets [16]. Meanwhile, they extended methods of solving fuzzy quadratic programming problems. In 2012, Molai provided sufficient conditions to determine the optimal solution of the fuzzy relational quadratic programming model by applying the maximum or minimal solution of the feasible region [17]. In 2014, Molai proposed studying the minimization problem of quadratic target function with inequality restriction of maximum product fuzzy relation [18].

Fuzziness of both variables and constraints: In 2012, Silva used a new dual method to solve quadratic programming problems with fuzzy coefficients and uncertain order relations in the constraint set [19]. In 2018, Coelho also proposed a novel dual method to solve these quadratic programming problems [20]. In addition, Gong (2009) gave first-order optimality conditions for fuzzy quadratic programming [21]. Kheirfam(2012) presented a strict sensitivity analysis of fuzzy quadratic programming [22]. Zhou (2014) provided the optimal condition of fuzzy quadratic programming [23], etc.

The problem of fully fuzz programming with fuzziness of variables and parameters is of universal significance. The existing literature of full fuzzy quadratic programming is as follows: In 2013, Kheirfam transformed such problems into deterministic models with nonlinear objectives and linear constraints through fuzzy ranking and algorithm operation [24]. In 2019, Mahajan et al. solved a quadratic programming problem comprising fuzzy parameters and fuzzy variables [25]. He proposed a direct method and KKT conditional method to solve it. In literature [24], the computation of the product of two triangular fuzzy numbers in the objective function is inaccurate, so the computation result is incorrect. In literature [25], there are six unknowns and six equations in constraint conditions of equality. According to the solution discrimination theorem of linear equations, the problem can be solved directly via the equations. KKT conditions are employed in literature to significantly increase the number of constraints, without achieving the effect of simplification.

Therefore, this paper proposes a new method to solve the full fuzzy quadratic programming problem. This method can be extended to fuzzy nonlinear programming problems. The basic structure of the paper is as follows: In Sect.2, we briefly present the applicable knowledge of triangular fuzzy numbers and suggest an enhanced fuzzy number ranking method. In Sect.3, we propose an A-PSO optimization algorithm to solve the problem. In Sect.4, we provide the model and solving steps of the fully fuzzy quadratic programming problem. In Sect.5, we provide the corresponding numerical examples. We apply the new ranking function and A-PSO optimization algorithm to the fully fuzzy quadratic programming problem. Finally, the effectiveness of the sorting function and algorithm is further illustrated.

II. Preliminaries

A. Triangular Fuzzy Number and Operations

The calculation of the triangular fuzzy numbers is straightforward. The expression is more in keeping with the way people think. It is merely essential to take the best value, the acceptable maximum and the minimum. Then a triangular fuzzy number can be represented. Therefore, in this chapter, we provide some fundamental theorems and conclusions about triangular fuzzy numbers and fuzzy sets.

Definition 1[26]**.** Suppose a mapping is given on the universe U : $A:U\rightarrow[0,1]$ and $u\rightarrow A(u)$. Then A is called a fuzzy set on U and $A(u)$ is membership function of A . For $\forall\lambda\in[0,1]$ has $A_\lambda=\{u\mid u\in U, A(u)\geq\lambda\}$, then A_λ is called a cut set of λ .

Definition 2[26]**.** Let R be real number field, $\tilde{A}\in F(R)$, if $\forall x_i\in R$, and $x_1>x_2>x_3$ has: $A(x_2)\geq A(x_1)\Lambda A(x_3)$, then $\tilde{A}$ is called a convex set F .

Definition 3[26]**.** Suppose $\tilde{A}$ is a normal set on the real number field R , $\forall\lambda\in(0,1]$, and A_λ are both a closed interval: $A_\lambda=[a_\lambda^l, a_\lambda^u]$, then $\tilde{A}$ is called an F real number (referred to as an F number). The totality of F numbers is denoted as $\tilde{R}$.

Definition 4[27]**.** A fuzzy number $\tilde{A}$, whose membership function is:

$$f_{\lambda(x)}=\begin{cases}(x-a^l)/(a^m-a^l), a^l\leq x\leq a^m\\ (a^u-x)/(a^u-a^m), a^m\leq x\leq a^u\\ 0, others\end{cases}$$

$\tilde{A}$ is called a triangular fuzzy number on $[a^l, a^u]$, and abbreviated as $\tilde{A}=(a^l,a^m,a^u)$. Assuming that fuzzy number has two fuzzy state values, which denotes $\tilde{A}=(a^l,a^{m_1},a^{m_2},a^u)$. λ -cut set of a triangular fuzzy number $\tilde{A}=(a^l,a^m,a^u)$ is $\tilde{A}_\lambda=[\tilde{A}_L(\lambda),\tilde{A}_R(\lambda)](0\leq\lambda\leq1)$. According to the membership function, the left endpoint $\tilde{A}_L(\lambda)=(a^m-a^l)\lambda+a^l$ and the right endpoint $\tilde{A}_R(\lambda)=a^u+(a^m-c^u)\lambda$ are obtained.

Definition 5[28]**.** When $a^l>0$, a triangular fuzzy number $\tilde{A}=(a^l,a^m,a^u)$ is positive; when $a^u<0$, a triangular fuzzy number $\tilde{A}=(a^l,a^m,a^u)$ is negative.

Definition 6[28]**.** Let the two triangular fuzzy numbers be $\tilde{A}_1=(a_1^l,a_1^m,a_1^u)$ and $\tilde{A}_2=(a_2^l,a_2^m,a_2^u)$, then:

(1) $\tilde{A}_1+\tilde{A}_2=(a_1^l+a_2^l, a_1^m+a_2^m, a_1^u+a_2^u)$

(2) $-\tilde{A}_1=(-a_1^u,-a_1^m,-a_1^l)$

(3) $\tilde{A}_1-\tilde{A}_2=(a_1^l-a_2^u, a_1^m-a_2^m, a_1^u-a_2^l)$

(4) $\tilde{A}_1\cdot\tilde{A}_2$

$$\tilde{A}_1\cdot\tilde{A}_2=\begin{cases}(a_1^la_2^l, a_1^ma_2^m, a_1^ua_2^u);(a_1^l\geq0, a_2^l\geq0)\\ (a_1^ua_2^l, a_1^ma_2^m, a_1^la_2^u);(a_1^l\geq0, a_2^u<0)\\ (a_1^la_2^u, a_1^ma_2^m, a_1^ua_2^l);(a_1^u<0, a_2^l\geq0)\\ (a_1^la_2^u, a_1^ma_2^m, a_1^ua_2^u);(a_1^l<0, a_1^u\geq0, a_2^u\geq0)\\ (a_1^ua_2^u, a_1^ma_2^m, a_1^la_2^l);(a_1^l<0, a_1^u\geq0, a_2^u<0)\end{cases}$$

Proof: In the product formula, only the last two formulas are proved in this paper. The proof of other formulas is shown in literature [25-28]. Triangular fuzzy numbers and LR fuzzy numbers can be converted to each other. The product formula of triangular fuzzy numbers can be extended by the product formula of LR fuzzy numbers. Let LR fuzzy numbers be $\tilde{M}_1=(m_1;\alpha_1,\beta_1)$ and $\tilde{M}_2=(m_2;\alpha_2,\beta_2)$. To literature [7], we have $\tilde{M}_1\cdot\tilde{M}_2=(m_1m_2;\alpha_1m_2+\alpha_1\beta_2-\beta_2m_1,\beta_2m_1+\beta_1\beta_2)$, when $m_1-\alpha_1<0, m_1+\beta_1\geq 0, \tilde{M}_2\geq 0$. We convert LR fuzzy numbers into triangular fuzzy numbers $a^l=m-\alpha$, $a^m=m$, and $a^u=m+\beta$. There is formula $\tilde{A}_1\cdot\tilde{A}_2=(a_1^l a_2^u, a_1^m a_2^m, a_1^u a_2^u)$, when $a_1^l<0$, $a_1^u\geq 0$, $a_2^l\geq 0$ are available. To the opposite of LR fuzzy number is $-(m;\alpha,\beta)_{LR}=-(m;\beta,\alpha)_{RL}$. There is a formula $\tilde{A}_1\cdot\tilde{A}_2=(a_1^u a_2^u, a_1^m a_2^m, a_1^l a_2^l)$ can be obtained, when $a_1^l<0$, $a_2^u<0$, $a_1^u\geq 0$ are available.

Definition 7[28]**.** If the two triangular fuzzy numbers $\tilde{A}_1=(a_1^l,a_1^m,a_1^u)$ and $\tilde{A}_2=(a_2^l,a_2^m,a_2^u)$ are equal, then:

$$a_1^l=a_2^l, a_1^m=a_2^m, a_1^u=a_1^u$$

B. *Improved Sorting Function*

Classifying fuzzy numbers is an important problem in solving fuzzy programming. At present, the classification problem of fuzzy numbers has been investigated by numerous scholars. The most typically used method is to transform fuzzy numbers into numbers with an existing order structure by mapping, which is also known as the ranking function method.

Definition 8[29]**.** Signed distance for triangular fuzzy numbers $\tilde{A}=(a^l,a^m,a^u)$:

$$d(\tilde{A})=\frac{1}{2}\int_0^1(\inf_{x\in\tilde{A}_\lambda}+\sup_{x\in\tilde{A}_\lambda})d\lambda=\frac{a^l+2a^m+a^u}{4} \quad (1)$$

This sorting index is the most widely used sorting method. According to the membership function, the ambiguity at different points in $\tilde{A}_\lambda$ is different. Therefore, we divide this region in two closed intervals: $[\tilde{A}_L(\lambda),a^m]$ and $[a^m,\tilde{A}_R(\lambda)]$. We take the value of fuzzy state as the cut-off point and propose a new sorting criterion.

Definition 9. The new ranking index for triangular fuzzy numbers $\tilde{A}=(a^l,a^m,a^u)$ is defined as follows:

$$R(\tilde{A})=\int_0^1[\frac{\tilde{A}_L(\lambda)+a^m}{2}\lambda]+[\frac{\tilde{A}_R(\lambda)+a^m}{2}\lambda]d\lambda \quad (2)$$
$$=\frac{(a^l+10a^m+a^u)}{12}$$

Among them, λ is the weight value of different interval segments. We give corresponding weight values according to different intervals. It's set to $(a^m-\tilde{A}_L(\lambda))/(a^m-a^l)=1-\lambda$. As λ increases, the ambiguity of each point increases, and the corresponding weight value should also increase. So, the weight ends up being $1-(1-\lambda)=\lambda$. Similarly, the weight of $[a^m,\tilde{A}_R(\lambda)]$ can be obtained as λ. Symmetrical triangular fuzzy numbers, such as $(1,2,3)$ and $(1.5,2,2.5)$, have the same sorting value when sorted by Formula 2. But the two images do not overlap completely through image comparison. To avoid this situation, this paper adds discrete computation to the new ranking criteria.

Definition 10[30]. Suppose the $\tilde{A}_i(a_i^l,a_i^{m_1},a_i^{m_2},a_i^u;\omega_i)$ $(i=1,...,n)$ is a set of LR-type fuzzy numbers. The left and right separation degree of fuzzy number $\tilde{A}_i$ is defined as:

$$S_{\tilde{A}_i}^R=\int_0^{\omega_i}[x_{\max}-\tilde{A}_i^R(\alpha)]d\alpha$$
$$S_{\tilde{A}_i}^L=\int_0^{\omega_i}[\tilde{A}_i^L(\alpha)-x_{\min}]d\alpha$$

Definition 11[30]**.** Suppose the $\tilde{A}_i(a_i^l,a_i^{m_1},a_i^{m_2},a_i^u;\omega_i)$ $(i=1,...,n)$ is a set of LR-type fuzzy numbers. I is the mapping from $\tilde{A}_i$ to the real number R, which denotes $I(\tilde{A}_i)=\alpha S_{\tilde{A}_i}^L-S_{\tilde{A}_i}^R+\alpha S_{\tilde{A}_i}^R$. Among them, α is preference coefficient of the decision maker. In this paper, ω is set to 1, and a fuzzy number ranking method based on the calculation of the degree of separation is proposed.

Definition 12. A new sorting function for triangular fuzzy numbers is defined by combining phase separation calculation and sorting criteria:

$$P(\tilde{A})=R(\tilde{A})+I(\tilde{A})=\frac{1}{12}(-3a^l+16a^m-a^u)$$
$$=\int_0^1\left[\frac{\tilde{A}(\lambda)+a^m}{2}\lambda\right]+\left[\frac{a^m+\overline{\tilde{A}}(\lambda)}{2}\lambda\right]d\lambda+\alpha S_{\tilde{A}_i}^L-(1-\alpha)S_{\tilde{A}_i}^R$$

Among them, $R(\tilde{A})$ is the redefined sorting criterion, and $I(\tilde{A})$ is the calculation formula of phase separation degree. The comparison rule of the sorting function is that the ambiguity number increases with the decrease of the left phase separation and the increase of the right phase separation. Thus, the improved trig function sorting function is obtained, and its sorting criterion is:

$$P(\tilde{A}_1)>P(\tilde{A}_2)\Rightarrow\tilde{A}_1>_P\tilde{A}_2$$
$$P(\tilde{A}_1)<P(\tilde{A}_2)\Rightarrow\tilde{A}_1<_P\tilde{A}_2$$

C. *Effectiveness Analysis*

The ranking function which proposed in this paper is compared with the classical ranking method of fuzzy numbers, such as Yager method [31], Roubens method [32] and S.Abbasbandy method [30]. Let $\tilde{A}_1=(1,2,3)$, $\tilde{A}_4=(0,2,4)$ $\tilde{A}_2=(1,2.5,4)$, $\tilde{A}_3=(0.5,2.5,4.5)$. Comparison results are shown in Table 1:

TABLE 1
COMPARISON RESULTS

Method	$\tilde{A}_1$	$\tilde{A}_2$	$\tilde{A}_3$	$\tilde{A}_4$	Sort results
Yager	2	2.5	2.5	2	$\tilde{A}_2=\tilde{A}_3>\tilde{A}_1=\tilde{A}_4$
Roubens	4	5	5	4	$\tilde{A}_2=\tilde{A}_3>\tilde{A}_1=\tilde{A}_4$
S.Abbasbandy	0.17	0.25	0.33	0.33	$\tilde{A}_3=\tilde{A}_4>\tilde{A}_2>\tilde{A}_1$
Method of This paper	**2.17**	**2.75**	**2.83**	**2.33**	$\tilde{A}_3>\tilde{A}_2>\tilde{A}_4>\tilde{A}_1$

As shown in Figure 1, the images of the four fuzzy numbers are not entirely overlapped. Therefore, it can be considered that these fuzzy numbers are utterly unequal. In Table 1, the three classical algorithms have two identical values, which are not distinguished. Four fuzzy numbers can be thoroughly compared with the method in this paper, and the results are reasonable.

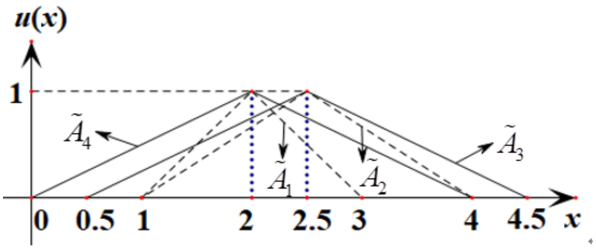

Fig. 1. Image of four fuzzy numbers

Definition 13[23]. Matrix $\tilde{Q}=(\tilde{q}_{kj})_{n\times n}$ is a symmetric fuzzy matrix and should satisfy $\tilde{q}_{ij}=\tilde{q}_{ji}$.

The fuzzy parameters can be transformed into a certain number through the ranking function, and $Q=R(\tilde{Q})$ can be obtained. According to the literature [23], the optimality conditions for fuzzy quadratic programming problems are summarized, as shown in Table 2:

TABLE 2
OPTIMALITY CONDITIONS

Condition	Result 1	Result 2
$R(\tilde{Q})$ is positive definite matrix	$\tilde{Q}$ is fuzzy positive definite matrix	Strictly convex programming problem
$R(\tilde{Q})$ is positive semidefinite matrix	$\tilde{Q}$ is fuzzy positive semidefinite matrix	Convex programming problem
$R(\tilde{Q})$ is negative definite matrix	$\tilde{Q}$ is fuzzy negative definite matrix	none
$R(\tilde{Q})$ is Semi negative definite matrix	$\tilde{Q}$ is fuzzy negative definite matrix	none

III. Accelerated Particle Swarm Optimization with Penalty Function

The two biggest problems of the PSO algorithm are slow convergence and easy to fall into local minimum. To enhance the convergence ability of basic particle swarm optimization [33-36], Eberhart and Kennedy introduced parameter inertia weight w in basic particle swarm optimization. The update equations for the particle's position and velocity are then:

$$v_i(t+1)=w\cdot v_i(t)+c_1\cdot rand_1\cdot(p_i-x_i(t)) \\ +c_2\cdot rand_2\cdot(p_g-x_i(t)) \tag{3}$$

$$x_i(t+1)=x_i(t)+v_i(t+1) \tag{4}$$

We divide the update equation into three parts. $w\cdot v_i(t)$ is the previous velocity of the particle. $c_1\cdot rand_1\cdot(p_i-x_i(t))$ is the thinking ability of particle itself. $c_2\cdot rand_2\cdot(p_g-x_i(t))$ is the position adjustment of the particle motion. As the search progresses, w, c_1 and c_2 are gradually decreasing in order to facilitate the convergence of the algorithm [34]. Therefore, the above update equation can be improved and expressed more concisely.

A. Accelerated Particle Swarm Optimization

According to the principle of particle swarm optimization, the values of w, c_1 and c_2 are gradually decreasing as a whole. Therefore, in the update process of particle swarm optimization, it can be replaced by decaying random numbers. The range is in the whole search interval. In the literature [35], a particle swarm algorithm based on random decay factor is proposed. According to the proof process, a new state transition equation can be finally obtained:

$$X_i=p_i+\alpha(p_g-p_i)+\beta'\cdot rand\cdot scale \tag{5}$$

α is related to the inertia parameter. $\beta'=\beta\cdot\gamma$ is the attenuation factor related to the learning factor. γ reflects the attenuation speed of β. In the improved equation, the two random numbers $rand_1$ and $rand_2$ are replaced by $\beta'\cdot rand$ to simplify the calculation and reduce the amount of calculation. β' increases randomness. As the parameter value increases, its randomness also increases. So, the value is not easy to be too large. Here take $\beta=0.2$. α is used to control the convergence speed when the particle adjusts its position. Too fast convergence speed is not conducive to searching other places. So, $\alpha=0.5$ is chosen. γ affects the search ability of particles. If the value is too low, the algorithm will easily fall into local optimization. Therefore, $\gamma=0.95$ is selected.

B. Penalty Function

In the optimization process, equality and inequality constraints need to be considered. In the calculation process, the required number of constraints need to be calculated according to the current parameters of each example. If the constraints are satisfied, the current particle value is taken. If not, the penalty term is added. The construction idea of the penalty function is to transform a constrained optimization problem into an unconstrained problem to solve it [37]. In optimization problem $<f,D>$, D is the feasible region satisfying constraint conditions, and $f:D\to R^n$ is the objective function. The solution of its minimum value is shown as follows:

$$\min \quad f(x) \quad x\in D$$
$$D=\{x\,|\,x\in R^n, h_i(x)\ge 0, i=1,...,m\}$$

Among them, $g_i(x)\ge 0$ as constraint conditions. The constraint $g_i(x)\ge 0$ is equivalent to the equality constraint $\min(g_i(x),0)=0$. Therefore, the inequality constraint problem can be transformed into the following equality constraint problem:

$$\min \quad f(x) \quad x\in R^n$$
$$\min(h_i(x),0)=0,(x\in R^n; i=1,...,m)$$

Let $g(x)=\sum_{i=1}^{m}(h_i(x),0)^2=0$. Then, the solution of the original problem can be transformed into the minimum probl em for solving unconstrained functions as shown below:

$$F(x,M)=f(x)+Mg(x)=f(x)+M\sum_{i=1}^{m}(h_i(x),0)^2$$

$F(x,M)$ is the penalty function. M is the penalty factor and constant. $Mg(x)$ is the penalty term. When M is sufficiently large, the optimal solution of $F(x,M)$ can approach the optimal solution of the constraint problem [34].

C. A-PSO Algorithm Process and Steps

Step1: According to the specific optimization problem, determine the particle dimension, give the particle location range, and set the relevant parameters: population size is $n=25$, $\alpha=0.5$, $\beta=0.2$, $\gamma=0.95$, the number of iterations is 150;

Step2: Initialize particle position and velocity, evaluate individual fitness of each particle in each population;

Step:3: Update individual value and global extreme value according to individual fitness value;

Step4: Update the position and velocity of each particle according to the improved formula (5);

Step5: If the end condition is satisfied, the algorithm ends. Otherwise, go to step2.

IV. FFQP Problem Solving based on Improved Sorting Function and A-PSO Algorithm

Fully fuzzy quadratic programming problem model (FFQP problem):

$$\begin{aligned}
&\min(or\max)\quad \tilde{z}=\tilde{c}^T\tilde{x}+\frac{1}{2}\tilde{x}\tilde{Q}\tilde{x}\\
&(resp.\tilde{c}_j^T\tilde{x}_j+\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{x}_i\tilde{q}_{ij}\tilde{x}_j)\\
&\qquad \tilde{A}\tilde{x}(resp.\sum_{j=1}^{n}\tilde{a}_{ij}\tilde{x}_j)\leq,\geq\tilde{b}(\tilde{b}_i)\\
&s.t.\quad \tilde{G}\tilde{x}(resp.\sum_{j=1}^{n}\tilde{g}_{sj}\tilde{x}_j)=\tilde{h}(\tilde{h}_s)\\
&\qquad \tilde{x}(resp.\tilde{x}_j)\geq 0
\end{aligned}$$

In the model above $\tilde{c}^T=[\tilde{c}_j^T]_{1\times n}$, $\tilde{x}=[\tilde{x}_j]_{n\times 1}$, $\tilde{Q}=[\tilde{q}_{ij}]_{n\times n}$, $\tilde{A}=[\tilde{a}_{ij}]_{m\times n}$, $\tilde{b}=[\tilde{b}_i]_{m\times 1}$, $\tilde{G}=[\tilde{g}_{sj}]_{m\times n}$, $\tilde{h}=[\tilde{h}_s]_{m\times 1}$. Among them, the elements of $\tilde{c}$, $\tilde{x}$, $\tilde{b}$, $\tilde{Q}$, $\tilde{A}$ and $\tilde{G}$ are all composed of triangular fuzzy numbers, and $\tilde{Q}$ is a fuzzy symmetric matrix. Therefore, we let $\tilde{c}_j^T=(c_j^l,c_j^m,c_j^u)^T$, $\tilde{x}_j=(x_j^l,x_j^m,x_j^u)$, $\tilde{q}_{ij}=(q_{ij}^l,q_{ij}^m,q_{ij}^u)$, $\tilde{a}_{ij}=(a_{ij}^l,a_{ij}^m,a_{ij}^u)$ $\tilde{b}_j=(b_j^l,b_j^m,b_j^u)$, $\tilde{g}_{sj}=(g_{sj}^l,g_{sj}^m,g_{sj}^u)$ and $\tilde{h}_s=(h_s^l,h_s^m,h_s^u)$ be triangular fuzzy numbers.

The solving steps of the FFQP problem are as follows:

Step1: Given by definition 5.

$\sum_{j=1}^{n}\tilde{c}_j^T\tilde{x}_j=\sum_{j=1}^{n}(c_j^l,c_j^m,c_j^u)^T\cdot(x_j^l,x_j^m,x_j^u)=(d_{ij}^l,d_{ij}^m,d_{ij}^c)$

$\sum_{j=1}^{n}\tilde{a}_{ij}\tilde{x}_j=(n_{ij}^l,n_{ij}^m,n_{ij}^u)$; $\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{x}_i\tilde{q}_{ij}\tilde{x}_j=(k_{ij}^l,k_{ij}^m,k_{ij}^u)$; $\sum_{s=1}^{n}\tilde{g}_{sj}\tilde{x}_j=(e_{sj}^l,e_{sj}^m,e_{sj}^u)$. Then the fully fuzzy quadratic programming problem can be transformed into:

$$\begin{aligned}
&\min(or\max)\quad \tilde{z}=(d_{ij}^l,d_{ij}^m,d_{ij}^c)+(k_{ij}^l,k_{ij}^m,k_{ij}^u)\\
&\qquad (n_{ij}^l,n_{ij}^m,n_{ij}^u)\geq,\leq(b_j^l,b_j^m,b_j^u)\\
&s.t.\qquad e_{sj}^l=h_s^l,e_{sj}^m=h_s^m,e_{sj}^u=h_s^u\\
&\qquad (x_j^l,x_j^m,x_j^u)\geq 0
\end{aligned}$$

Step2: The sorting function is transformed into a clear model.

Uncertain quadratic programming can be expressed in many forms [38-39]. This paper adopts the above model. According to Formula 2, the full fuzzy model in Step1 is transformed into a clear quadratic programming model. The model is as follows:

$$\begin{aligned}
&\min(or\max)\quad z=\frac{1}{12}\begin{bmatrix}(-3d_{ij}^l+16d_{ij}^m-d_{ij}^u)\\+(-3k_{ij}^l+16k_{ij}^m-k_{ij}^u)\end{bmatrix}\\
&\qquad(-3n_{ij}^l+16n_{ij}^m-n_{ij}^u)\geq,\leq(-3b_j^l+16b_j^m-b_j^u)\\
&s.t.\quad e_{sj}^l=h_s^l,e_{sj}^m=h_s^m,e_{sj}^u=h_s^u\\
&\qquad x_j^l\geq 0,x_j^m-x_j^l\geq 0,x_j^u-x_j^m\geq 0
\end{aligned}$$

Step3: The A-PSO algorithm is used to solve the clear programming problem in Step4 and the value of $\tilde{x}_j=(x_j^l,x_j^m,x_j^u)$ is obtained.

Step4: Substitute the value of $\tilde{x}_j=(x_j^l,x_j^m,x_j^u)$ obtained by Step3 into the objective function as shown in formula (6). We need can be obtained the target value $\tilde{z}=(z^l,z^m,z^u)$.

$$\tilde{z}=\sum_{j=1}^{n}\tilde{c}_j^T\tilde{x}_j+\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\tilde{x}_i\tilde{q}_{ij}\tilde{x}_j \tag{6}$$

V. Numerical Examples

A. Examples from the Literature

(1) Example of inequality constraints:

This calculation example is selected from literature [24]. In Step2, the product operation of triangular fuzzy numbers is used, and the result is wrong. So, it is revised. The result of its revision is shown in the result given by Step1 below.

$$\begin{aligned}
&\min\quad(-3,-2,-1)\cdot(x_1,y_1,z_1)\\
&+(-9,-6,-3)\cdot(x_2,y_2,z_2)\\
&+\frac{1}{2}((x_1,y_1,z_1),(x_2,y_2,z_2))\\
&\begin{pmatrix}(0,1,2) & (-4,-2,4)\\(-4,-2,4) & (-4,3,6)\end{pmatrix}\cdot\begin{pmatrix}(x_1,y_1,z_1)\\(x_2,y_2,z_2)\end{pmatrix}\\
&\qquad(0,1,2)\cdot(x_1,y_1,z_1)\\
&\qquad+(-1,1,3)\cdot(x_2,y_2,z_2)\leq(-2,3,4)\\
&s.t.\quad(-4,-2,4)\cdot(x_1,y_1,z_1)\\
&\qquad+(1,2,3)\cdot(x_2,y_2,z_2)\leq(-2,3,4)\\
&\qquad(0,2,4)\cdot(x_1,y_1,z_1)+(x_2,y_2,z_2)\leq(1,3,5)\\
&\qquad(x_i,y_i,z_i)\geq 0,i=1,2
\end{aligned}$$

Existence of solution:

$R(\tilde{Q})=\begin{pmatrix}(0,1,2) & (-4,-2,4)\\(-4,-2,4) & (-4,3,6)\end{pmatrix}=\begin{pmatrix}1.17 & -2\\-2 & 4.5\end{pmatrix}$ is a positive definite matrix. Then, $\tilde{Q}$ is a fuzzy positive definite matrix. Therefore, the problem is a convex programming problem. There is a unique optimal solution.

Step1: According to the four operations of triangular fuzzy number, the following model can be obtained:

$$\min \quad \begin{aligned}&(-3z_1-9z_2,-2y_1-6y_2,-x_1-3x_2)\\&+(-4z_1z_2-2z_2^2,0.5y_1^2-2y_1y_2\\&+1.5y_2^2,z_1^2+4z_1z_2+3z_2^2)\end{aligned}$$

$$s.t.\quad \begin{aligned}&(-z_2,y_1+y_2,2z_1+3z_2)\le(-2,3,4)\\&(-4z_1+x_2,-2y_1+2y_2,4z_1+3z_2)\le(-2,3,4)\\&(x_2,2y_1+y_2,4z_1+z_2)\le(1,3,5)\\&(x_i,y_i,z_i)\ge 0,i=1,2\end{aligned}$$

Step2: The standard quadratic programming model can be obtained through the improved sorting function:

$$\min \quad \frac{1}{12}(9z_1+27z_2-32y_1-96y_2+x_1+3x_2 +8y_1^2-32y_1y_2+24y_2^2-z_1^2)$$

$$s.t.\quad \begin{aligned}&16y_1+16y_2-2z_1\le 50\\&8z_1-3x_2-32y_1+32y_2-3z_2\le 50\\&-3x_2+32y_1+16y_2-4z_1-z_2\le 40\\&x_1\ge 0,y_1-x_1\ge 0,z_1-y_1\ge 0,\\&x_2\ge 0,y_2-x_2\ge 0,z_2-y_2\ge 0\end{aligned}$$

Step3: The quadratic programming model is solved by A-PSO algorithm.

Results analysis:

TABLE 3
RESULTS OF INEQUALITY CONSTRAINT EXAMPLE

	x_1	y_1	z_1	x_2	y_2	z_2	Function value
Original literature	0.6462	0.6462	0.6462	1.2308	1.2308	1.4769	-7.8747
A-PSO	**1.0458**	**3.4459**	**18.8876**	**0.2332**	**0.3359**	**0.3359**	**-15.9931**

The A-PSO algorithm is used to solve the problem. The solution results are $\tilde{x}_1=(1.0458,3.4459,18.8876)$ and $\tilde{x}_2=(0.2332,0.3359,0.3359)$, according to the Table 3. The function value obtained by A-PSO algorithm is -15.9931, which is better than the result given in the original literature -7.8747. Meanwhile, A-PSO algorithm was compared with GA and PSO algorithm, as shown in Table 4.

Methods to analyze:

TABLE 4
RESULT VALUES OF DIFFERENT METHODS

	x_1	y_1	z_1	x_2	y_2	z_2	Function value
GA	1.4006	3.6936	18.9571	0.0324	0.1053	0.1242	-13.3797
PSO	1.8000	3.0000	17.1663	0.0100	0.3000	0.3000	-13.0083
A-PSO	**1.0458**	**3.4459**	**18.8876**	**0.2332**	**0.3359**	**0.3359**	**-15.9931**

In Table 4, GA and PSO algorithms are easy to fall into local optimization, which the results are -13.3797 and -13.0083 respectively. The results of the GA algorithm are better than that of the PSO algorithm. Overall, the calculation result of the A-PSO is -9.4742, which is the best.

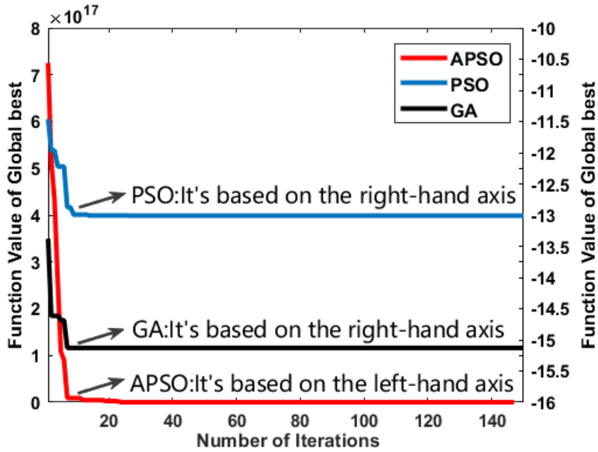

Fig. 2. Fitness curves of inequality constrained examples

Meanwhile, fitness curves of A-PSO, GA and PSO algorithms are presented, as shown in Figure 2.

Step4: Substitute the solution result into the objective function, and obtain $\tilde{z}=(-85.2889,-5.1158,380.7119)$

(2) Example of equation constraint:

This calculation example is selected from literature [25]. It calculates two equations with six equations to solve six unknowns. According to the solution discrimination theorem for linear equations, this problem can be solved directly via equation. There is no need to use sophisticated methods to solve it. Therefore, to avoid such a situation, in this paper we consider merging two equations into one by four operations on trigonometric fuzzy numbers. The exact procedure is as follows.

$$\min \quad \begin{aligned}&(2,3,4)\cdot(x_1,y_1,z_1)\\&+(-5,-4,-3)\cdot(x_2,y_2,z_2)\\&+\frac{1}{2}((x_1,y_1,z_1),(x_2,y_2,z_2))\\&\cdot\begin{pmatrix}(8,10,12) & (0,2,4)\\(0,2,4) & (2,4,6)\end{pmatrix}\cdot\begin{pmatrix}(x_1,y_1,z_1)\\(x_2,y_2,z_2)\end{pmatrix}\end{aligned}$$

$$s.t.\quad \begin{aligned}&(1,2,3)\cdot(x_1,y_1,z_1)+(-1,0,1)\cdot(x_2,y_2,z_2)\\&=(-0.75,0,0.75)\\&(1,2,3)\cdot(x_1,y_1,z_1)+(2,3,4)\cdot(x_2,y_2,z_2)\\&=(1,2,3)\\&(x_i,y_i,z_i)\ge 0,i=1,2\end{aligned}$$

Existence of solution:

$R(\tilde{Q})=\begin{pmatrix}(8,10,12) & -(0,1,2)\\-(0,1,2) & (2,4,6)\end{pmatrix}=\begin{pmatrix}10.33 & -1.17\\-1.17 & 4.33\end{pmatrix}$ is a positive definite matrix. Then, $\tilde{Q}$ is a fuzzy positive definite matrix. Therefore, the problem is a convex programming problem. There is a unique optimal solution.

Step1: According to the four operations of triangular fuzzy number, the following model can be obtained:

$$\min \quad \begin{aligned}&(2x_1-5z_2,3y_1-4y_2,4z_1-3x_2)\\&+(4x_1^2-2z_1z_2+x_2^2,5y_1^2-y_1y_2+2y_2^2,6z_1^2+3z_2^2)\end{aligned}$$

$$s.t.\quad \begin{aligned}&(x_1-z_2-2z_1+2x_2/y_1+3y_2/\\&3z_1+5z_2)=(0.25,2,3.75)\\&(x_i,y_i,z_i)\ge 0,i=1,2\end{aligned}$$

Step2: The standard quadratic programming model can be obtained through the improved sorting function:

$$\min \quad \frac{1}{12}\left(\begin{aligned}&-6x_1+15z_2-12x_1^2+6z_1z_2-3x_2^2\\&+48y_1-64y_2+80y_1^2-16y_1y_2+32y_2^2\\&-4z_1+3x_2-6z_1^2-3z_2^2\end{aligned}\right)$$

$$s.t.\quad \begin{aligned}&x_1-z_2-2z_1+2x_2=0.25,\\&y_1+3y_2=2, 3z_1+5z_2=3.75,\\&x_1\geq 0, y_1-x_1\geq 0, z_1-y_1\geq 0,\\&x_2\geq 0, y_2-x_2\geq 0, z_2-y_2\geq 0\end{aligned}$$

Step3: The quadratic programming model is solved by A-PSO algorithm.

Results analysis:

TABLE 5
RESULTS OF EQUALITY CONSTRAINT EXAMPLE

	x_1	y_1	z_1	x_2	y_2	z_2	Function value
Original literature	0	0	0	0.5000	0.6660	0.7500	-1.2947
A-PSO	**0**	**0**	**0**	**0.4969**	**0.6668**	**0.7506**	**-1.5154**

According to Table 5, the results are $\tilde{x}_1=(0,0,0)$ and $\tilde{x}_2=(0.4969,0.6668,0.7506)$. The function value obtained by A-PSO algorithm is -1.5152, which is better than the result given in the original text, -1.2947. At the same time, different algorithms were compared, and the solution results were compared with the original results, as shown in Table 6.

Methods to analyze :

TABLE 6
RESULT VALUES OF DIFFERENT METHODS

	x_1	y_1	z_1	x_2	y_2	z_2	Function value
GA	0	0	1	0.5139	0.6412	0.7217	-1.9405
PSO	0	0	0	0.5000	0.6667	0.7500	-1.5073
A-PSO	**0**	**0**	**0**	**0.4969**	**0.6668**	**0.7506**	**-1.5154**

The algorithm result pairs are shown in Table 6. Result obtained by GA is -1.9405, but it does not satisfy the equation constraint. Calculation result of PSO algorithm is -1.5073, which is limited to the value range of x_i in the equality constraints. A-PSO algorithm satisfies the equality constraint and is not limited by parameters, and its calculation result is the best, which is -1.5154.

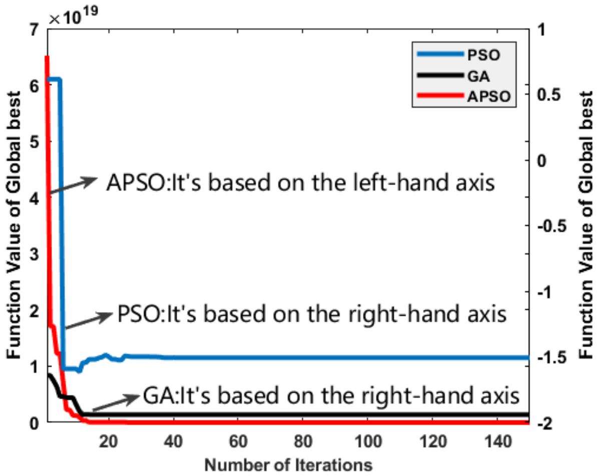

Fig. 3. Fitness curve of an equation constrained example

The fitness curves of A-PSO, GA and PSO algorithms in the equation constraint example are shown in Figure 3.

Step4: Substitute the solution result into the objective function and obtain $\tilde{z}=(-3.5061,-1.7780,0.1995)$.

The results and method analysis of the above inequality and equality constraint examples show that the new sorting function and A-PSO algorithm proposed in this paper are effective. And the results obtained are better.

B. Full Fuzzy Nonlinear Programming Problem with Equality and Inequality Constraints

We extend this method to solve fully fuzzy nonlinear programming problems, as shown below.

Example:

$$\begin{aligned}\min\quad &(9,10,11)\tilde{x}_1^2+(7.5,10,11)\tilde{x}_2^2-(3.5,5,8)\tilde{x}_3^2\\&-(4,5,7)\tilde{x}_1-(47,50,52)\tilde{x}_2-(9,10,11)\tilde{x}_3\end{aligned}$$

$$s.t.\quad\begin{aligned}&\tilde{x}_1^2+\tilde{x}_2^2+\tilde{x}_3^2-\tilde{x}_1-\tilde{x}_2-\tilde{x}_3\leq(65,75,80)\\&(0.7,1,1.3)\tilde{x}_1^2+(1,2,3)\tilde{x}_2^2\\&+(1.4,2,2.2)\tilde{x}_3^2-\tilde{x}_1-\tilde{x}_2\leq(14,39,64)\\&(1,2,2.2)\tilde{x}_1^2+(0.3,1,1.5)\tilde{x}_2^2\\&+(1.4,2,2.2)\tilde{x}_1+\tilde{x}_2+(0.5,0.7,1)\tilde{x}_3=(4,15,40)\\&(x_i,y_i,z_i)\geq 0, i=1,2,3\end{aligned}$$

Step1: According to the four operations of triangular fuzzy number, the following model can be obtained:

$$\min\quad\left(\begin{aligned}&9x_1^2+7.5x_2^2+3.5x_3^2-7z_1-52z_2-11z_3\\&/10y_1^2+10y_2^2+5y_3^2-5y_1-50y_2-10y_3\\&/11z_1^2+11z_2^2+8z_3^2-4x_1-47x_2-9x_3\end{aligned}\right)$$

$$s.t.\quad\begin{aligned}&(x_1^2+x_2^2+x_3^2-z_1-z_2-z_3/\\&y_1^2+y_2^2+y_3^2-y_1-y_2-y_3/\\&z_1^2+z_2^2+z_3^2-x_1-x_2-x_3)\leq(65,75,80)\\&(0.7x_1^2+x_2^2+1.4x_3^2-z_1-z_3/\\&y_1^2+2y_2^2+2y_3^2-y_1-y_3/\\&1.3z_1^2+3z_2^2+2.2z_3^2-x_1-x_3)\leq(14,39,64)\end{aligned}$$

$$s.t.\quad\begin{aligned}&x_1^2+0.3x_2^2+1.4x_1+x_2+0.5x_3=4\\&2y_1^2+y_2^2+2y_1+y_2+0.7y_3=15\\&2.2z_1^2+1.5z_2^2+2.2z_1+z_2+z_3=40\\&(x_i,y_i,z_i)\geq 0, i=1,2,3\end{aligned}$$

Step2: The nonlinear programming model is obtained by the improved sorting function:

$$\min\quad\frac{1}{12}\left(\begin{aligned}&-27x_1^2-22.5x_2^2-10.5x_3^2+21z_1+156z_2\\&+33z_3+160y_1^2+160y_2^2+80y_3^2-80y_1-800y_2\\&-160y_3-11z_1^2-11z_2^2-8z_3^2+4x_1+47x_2+9x_3\end{aligned}\right)$$

$$\begin{aligned}&(-3x_1^2-3x_2^2-3x_3^2+3z_1+3z_2+3z_3+\\&16y_1^2+16y_2^2+16y_3^2-16y_1-16y_2-16y_3\\&-z_1^2-z_2^2-z_3^2+x_1+x_2+x_3)\leq 925\\&(-2.1x_1^2-3x_2^2-4.2x_3^2+3z_1+3z_3\\&+16y_1^2+32y_2^2+32y_3^2-16y_1-16y_3\\&-1.3z_1^2-3z_2^2-2.2z_3^2+x_1+x_3)\leq 518\end{aligned}$$

$$s.t.\quad \begin{aligned} & x_1^2+0.3x_2^2+1.4x_1+x_2+0.5x_3=4 \\ & 2y_1^2+y_2^2+2y_1+y_2+0.7y_3=15 \\ & 2.2z_1^2+1.5z_2^2+2.2z_1+z_2+z_3=40 \\ & x_i\geq 0, y_i-x_i\geq 0, z_i-y_i\geq 0, i=1,2,3 \end{aligned}$$

Step3: Nonlinear programming is solved by A-PSO .

***Results analysis*:**

TABLE 7:
RESULTS OF EQUALITY AND INEQUALITY CONSTRAINTS EXAMPLE

	x_1	y_1	z_1	x_2	y_2	z_2	x_3	y_3	z_3
PSO	1.000	1.600	1.700	0.100	2.500	3.040	0.100	1.500	13.000
A-PSO	**0.318**	**0.956**	**1.016**	**2.011**	**2.720**	**3.515**	**0.467**	**1.614**	**13.451**

The fuzzy nonlinear programming problem can also be transformed into a clear nonlinear programming problem by the four rules of fuzzy number operation and sorting function. Results of A-PSO algorithm are $\tilde{x}_1=(0.318,0.956,1.016)$, $\tilde{x}_2=(2.011,2.720,3.515)$, $\tilde{x}_3=(0.467,1.614,13.451)$. Results of PSO are $\tilde{x}_1=(1,1.6,1.7)$, $\tilde{x}_2=(0.1,2.5,3.04)$ and $\tilde{x}_3=(0.1,1.5,13)$. Function value of A-PSO algorithm is -112.524, which is better than the function value of PSO algorithm is -129.748.

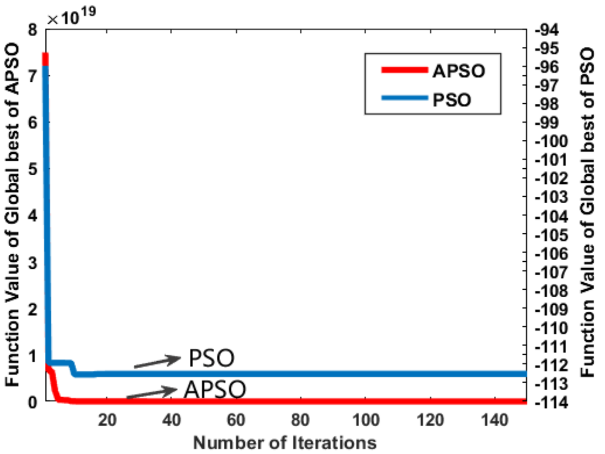

Fig. 4. Fitness curve of the example

The fit curves of the two algorithms are shown in Figure 4. In a more intricate nonlinear programming model, it can be seen that the A-PSO algorithm demonstrates good robustness. The host system can converge rapidly and the optimal results are more consistent with the problem. Therefore, the synthesis of these three examples displays the effectiveness of the A-PSO algorithm and the remarkable efficiency of optimization. It is shown that the proposed method can be expanded to fuzzy nonlinear programming problems and good results can be obtained.

Step4: Substitute the solution result into the objective function, and obtain

$$\tilde{z}=(-305.8487,-60.7717,1506.7715)\ .$$

VI. CONCLUSION

In this paper, the solution method of the fully fuzzy quadratic programming problem is discussed. In particular, we extend the four operations of triangular fuzzy numbers and propose a novel ranking method of fuzzy numbers. Meanwhile, the specific solution steps of the fully fuzzy quadratic programming are given, as follows: combining the four operations, diversification by a sorting function, solving the problem with the A-PSO algorithm, and putting the solution results into objective function values. Numerical calculation is performed for examples in existing literature. Numerical examples are awarded to analyze the algorithm and the results, which show the efficiency of the proposed method. Finally, we extend the method to solve the fully fuzzy nonlinear programming problem, and the method is also effective.

REFERENCES

[1] Qian Songdi et al, “Operations Research 4th edition,” Tsinghua University Press, pp1-10, 2012.

[2] Bellman RE and Zadeh LA, “Decision making in a fuzzy environment,” Management Science, vol.17, pp141-164, 1970.

[3] Kaur J, Kumar A, “Mehar's method for solving fully fuzzy linear programming problems with L-R fuzzy parameters,” Applied Mathematical Modelling, vol.37, no.12-13, pp7142-7153, 2013.

[4] M. Ranjbar and S. Effati, “Symmetric and Right-Hand-Side Hesitant Fuzzy Linear Programming,” IEEE Transactions on Fuzzy Systems, vol.28, no.2, pp215-227, 2020.

[5] Boris Pérez-Caedo, Verdegay J L, Concepcion-Morales E R, et al, “Lexicographic Methods for Fuzzy Linear Programming,” 2020.

[6] Mojtaba Borza, “A New Method to Solve Multi-Objective Linear Fractional Problems,” Fuzzy Information and Engineering, vol.13, no.3, pp323, 2021.

[7] Bogdana Stanojević, Milan Stanojević, “Empirical Versus Analytical Solutions to Full Fuzzy Linear Programming,” Intelligent Methods in Computing, Communications and Control, vol.1243, pp220-233, 2020.

[8] Neha Gupta, Abdul Bari, “Fuzzy multi-objective optimization for optimum allocation in multivariate stratified sampling with quadratic cost and parabolic fuzzy numbers,” Journal of Statistical Computation and Simulation, vol.87, no.12, pp2372-2383, 2017.

[9] A. T. Ubando, K. D. T. Aguilar, “Fuzzy quadratic programming model for the optimal design of an algal bioenergy park under optimal price markdown percentage,” 2016 IEEE Region 10 Conference, pp936-941, 2016.

[10] Wu X L, Liu Y K, “Optimizing fuzzy portfolio selection problems by parametric quadratic programming,” Fuzzy Optimization and Decision Making, vol.11, no.4, pp411-449, 2012.

[11] Zhang X, Zhang W G, Cai R, “Portfolio adjusting optimization under credibility measures,” Journal of Computational and Applied Mathematics, 2010.

[12] Seyedeh Maedeh Mirmohseni, Seyed Hadi Nasseri, “A Quadratic Programming with Triangular Fuzzy Numbers,” Journal of Applied Mathematics and Physics, vol.5, no.11, pp2218-2227, 2017.

[13] Reza Ghanbari and Khatere Ghorbani-Moghadam, “Solving fuzzy quadratic programming problems based on ABS algorithm,” Soft Computing, vol.23, no.22, pp11343-11349, 2019.

[14] Reza Ghanbari, Khatere Ghorbani-Moghadam and Nezam Mahdavi-Amiri, “A variables neighborhood search algorithm for solving fuzzy quadratic programming problems using modified Kerre’s method,” Soft Computing, vol.23, no.23, pp12305-12315, 2019.

[15] Al-Mumtazah N S, Surono S, “Quadratic Form Optimization with Fuzzy Number Parameters: Multiobjective Approaches,” International Journal of Fuzzy Systems, vol.22, no.4, pp1191-1197, 2020.

[16] Cruz C, Silva R C, Verdegay J L, “Extending and relating different approaches for solving fuzzy quadratic problems” Fuzzy Optimization & Decision Making, vol.10, no.3, pp193-210, 2011.

[17] Molai A A, “The quadratic programming problem with fuzzy relation inequality constraints,” Computers & Industrial Engineering, vol.62, no.1, pp256-263, 2012.

[18] Molai A A, “A new algorithm for resolution of the quadratic programming problem with fuzzy relation inequality constraints,” Computers & Industrial Engineering, vol.72, no.6, pp306-314, 2014.

[19] Silva R C, Yamakami A, “Using ranking function in dual approach to solve quadratic programming problem under fuzzy environment,” Fuzzy Information Processing Society, IEEE, 2012.
[20] Coelho R, “Solving Real-World Fuzzy Quadratic Programming Problems by Dual Parametric Approach,” North American Fuzzy Information Processing Society Annual Conference, 2018.
[21] Gong L, Zhou X, Liu W, et al, “First-order Optimality Conditions for Fuzzy Number Quadratic Programming with Fuzzy Coefficients,” International Conference on Fuzzy Systems & Knowledge Discovery, IEEE, 2009.
[22] Kheirfam B, Verdegay J L, “Strict sensitivity analysis in fuzzy quadratic programming,” Fuzzy Sets & Systems, vol.198, pp99-111, 2012.
[23] Zhou X G, Cao B Y, Nasseri S H, “Optimality Conditions for Fuzzy Number Quadratic Programming with Fuzzy Coefficients,” Journal of Applied Mathematics, vol.2014, pp1-8, 2014.
[24] Kheirfam B, “A Method for solving fully fuzzy quadratic programming problems,” Acta Univ.Apulensis Math.Inform, vol.27, pp69-76, 2011.
[25] Mahajan, Sumati, Gupta, S.K, “Methods to Solve QPPs with Fuzzy Parameters and Fuzzy Variables,” Journal of intelligent & fuzzy systems: Applications in Engineering and Technology, pp2757-2767, 2019.
[26] Yang Ruibiao, Gao Yingyi, “Principle and Application of Fuzzy Mathematics,” South China University of Technology Press, pp3-111, 2006.
[27] Zimmermannhj, “Fuzzy set theory and its application,” Boston: Kluwer-Nijhoff, 1996.
[28] Didier Dubois, Henri Prade, “Fuzzy Sets and Systems: Theory and Applications,” Academic Press, pp9-58, 1980.
[29] Fortemps P, Roubens M, “Ranking and defuzzification methods based on area compensation,” Fuzzy Sets and Systems, vol.82, no.3, pp 319-330, 1996.
[30] S. Abbasbandy, T. Hajjari, “A new approach for ranking of trapezoidal fuzzy numbers,” Computers & Mathematics with Applications, vol.7, no.3, pp413-419, 2009.
[31] Yager R R, “Ranking fuzzy subsets over the unit interval,” IEEE Conference on Decision & Control Including the Symposium on Adaptive Processes IEEE, pp1435-1437, 1979.
[32] Fortemps P, Roubens M, “Ranking and defuzzification methods based on area compensation,” Fuzzy Sets and Systems, vol.82, no.3, pp 319-330, 1996.
[33] R.C. Eberhart, J. Kennedy, “A new optimizer using particle swarm theory,” Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya Japan, pp39-43, 1995.
[34] Yu Shengwei, “Matlab optimization algorithm case analysis and application,” Tsinghua University Press, pp170-178, 2014.
[35] Sun Shuqin, WU Chenyue, YAN Wenli, Li Mingnan, LIU Yujie, Yang Bohua, “Optimal power flow calculation method based on random attenuation factor particle swarm optimization,” Power System Protection and Control, vol.49, no.13, pp43-52, 2021.
[36] Hiroshi Sho, “Investigation of Particle Multi-Swarm Optimization with Diversive Curiosity,” Engineering Letters, vol.28, no.3, pp960-969, 2020.
[37] Bingzhuang Liu, “A New Family of Smoothing Exact Penalty Functions for the Constrained Optimization Problem,” Engineering Letters, vol.29, no.3, pp984-989, 2021.
[38] Xue Deng, Yingxian Lin, and Huidan Zhuang, “Uncertain Portfolio with Fuzzy Investment Proportion Based on Possibilistic Theory,” Engineering Letters, vol.29, no.2, pp803-812, 2021.
[39] Chunfeng Wang, Yaping Deng, and Peiping Shen, “A Global Optimization Algorithm for Solving Indefinite Quadratic Programming,” Engineering Letters, vol.28, no.4, pp1058-1062, 2020.

Xianfeng Ding, born in January 1975, graduated from Southwest Petroleum University with a master's degree in Applied Mathematics and a doctor's degree in petroleum engineering computing technology. He served as an associate professor of the teaching and Research Office of mathematics and applied mathematics and a master's supervisor of mathematics at Southwest Petroleum University.