Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 9897153, 13 pages
https://doi.org/10.1155/2017/9897153

Research Article

MOQPSO-D/S for Air and Missile Defense WTA Problem under Uncertainty

Hao Xu, Qinghua Xing, and Zhenhao Tian

School of Air and Missile Defense, Air Force Engineering University, Xi'an 710051, China

Correspondence should be addressed to Hao Xu; kgdxuhao@163.com

Received 16 December 2016; Revised 5 November 2017; Accepted 12 November 2017; Published 14 December 2017

Academic Editor: Erik Cuevas

Aiming at the shortcomings of single objective optimization for solving weapon target assignment (WTA) and the existing multiobjective optimization based WTA method having problems being applied in air and missile defense combat under uncertainty, a fuzzy multiobjective programming based WTA method was proposed to enhance the adaptability of WTA decision to the changes of battlefield situation. Firstly, a multiobjective quantum-behaved particle swarm optimization with double/single-well (MOQPSO-D/S) algorithm was proposed by adopting the double/single-well based position update method, the hybrid random mutation method, and the two-stage based guider particles selection method. Secondly, a fuzzy multiobjective programming WTA model was constructed with consideration of air and missile defense combat's characteristics. And, the uncertain WTA model was equivalently clarified based on the necessity degree principle of uncertainty theory. Thirdly, with particles encoding and illegal particles adjusting, the MOQPSO-D/S algorithm was adopted to solve the fuzzy multiobjective programming based WTA model. Finally, example simulation was conducted, and the result shows that the WTA model constructed is rational and MOQPSO-D/S algorithm is efficient.

1. Introduction

In information warfare, air attack is a highly integrated operation form. Defenders need to carry out the task of anti-aerodynamic targets and antiballistic missiles simultaneously [1]. As one of the key phases in air and missile defense combat, weapon target assignment (WTA) plays a significant role on operational effectiveness. Therefore, study on air and missile defense WTA problem is of great significance.

Given the importance, WTA has been studied with application of Lagrange relaxation algorithm [2], ant colony algorithm [3], genetic algorithm [4], clone selection algorithm [5], and particle swarm optimization algorithm [6]. Most of these researches construct WTA model based on single objective optimization with the objective function of maximizing the kill probability to enemy. They need to allocate all the firepower without consideration of the sequent operation need and cannot fit with the operational practice. In order to avoid allocating all firepower, a WTA model was constructed with the objective function of maximizing effectiveness-cost ratio in literature [6]. However, the adaptability of WTA decision, based on this method, to the changes of battlefield situation is still weak. Because this method is still a single objective optimization method, when conducting this method, the decision-making preferences need to be provided beforehand and only one WTA alternative can be given. Aiming at overcoming these weak points, the multiobjective optimization method was introduced into studying WTA problem [7–9]. With consideration of every objective, the Pareto solution set, obtained by multiobjective optimization method, can provide a series of WTA alternatives for the commander. And the final WTA decision will be made by choosing one from the alternatives. These studies [7–9] could provide useful reference for research on air and missile defense WTA problem. However, they cannot be applied to air and missile defense combat, which has the characteristic of multiweapon composite disposition, directly. Moreover, with the growing complexity of battlefield, the certain WTA method [3–9] can hardly be applied to air and missile defense combat under uncertainty.

In order to address the issues above, the WTA model for air and missile defense with uncertain threat information is constructed based on fuzzy multiobjective programming, with consideration of the air and missile defense combat's characteristics. Due to the superior performance of quantum-behaved particle swarm optimization algorithm, a multiobjective quantum-behaved particle swarm optimization with double/single-well algorithm is proposed for solving the WTA model.

2. Basic Theory

2.1. Concepts of Multiobjective Optimization

Definition 1 (multiobjective optimization problem). Taking the minimizing problem as an example, a multiobjective optimization problem can be defined as follows:

$$\begin{aligned} \min \quad & f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_o(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in \Omega. \end{aligned} \tag{1}$$

In (1), $f_l(\mathbf{x})$ $(l = 1, 2, \ldots, o)$ is objective function, $\mathbf{x}$ is the decision vector, and Ω is the decision space defined by a series of constraints.

Definition 2 (Pareto dominance). Let $\mathbf{x}_a$ and $\mathbf{x}_b$ be two feasible solutions in Ω; if

$$\text{all}(f_l(\mathbf{x}_a) \le f_l(\mathbf{x}_b)) \,\&\&\, \text{any}(f_l(\mathbf{x}_a) < f_l(\mathbf{x}_b)) \quad (l = 1, 2, \ldots, o), \tag{2}$$

then $\mathbf{x}_a$ Pareto dominates $\mathbf{x}_b$, denoted by $\mathbf{x}_a \prec \mathbf{x}_b$.

Definition 3 (Pareto optimal solution). Let $\mathbf{x}_a \in \Omega$; if $\neg\exists \mathbf{x}_b \in \Omega$, meeting $\mathbf{x}_b \prec \mathbf{x}_a$, then $\mathbf{x}_a$ would be defined as a Pareto optimal solution. And the set $\Omega_p = \{\mathbf{x}_a \in \Omega \mid \neg\exists \mathbf{x}_b \in \Omega, \mathbf{x}_b \prec \mathbf{x}_a\}$ would be defined as Pareto optimal solution set.

Definition 4 (Pareto front). The region which consisted of the objective values, corresponding to all solutions in Ω_p, is defined as Pareto front.

2.2. QPSO Algorithm. Quantum-behaved particle swarm optimization (QPSO) algorithm [10, 11], compared with the particle swarm optimization algorithm, has many advantages, such as faster convergence rate, fewer control parameters, and better global convergence. It has attracted much attention of scholars [12, 13].

QPSO algorithm is a quantum mechanics based optimization algorithm. It is assumed that the motion state of the particles in the optimal space relative to the attractor $\mathbf{P}$ can be described by the wave function $\psi(\mathbf{Y})$ in δ potential well. And $\psi(\mathbf{Y})$ is described as follows:

$$\psi(\mathbf{Y}) = \frac{1}{\sqrt{\mathbf{L}}} e^{-|\mathbf{Y}|/\mathbf{L}}. \tag{3}$$

In (3), $\mathbf{L}$ is the characteristic length of δ potential well.

The corresponding probability density function is $Q(\mathbf{Y}) = (1/\mathbf{L})e^{-2|\mathbf{Y}|/\mathbf{L}}$, and the probability distribution function is $F(\mathbf{Y}) = e^{-2|\mathbf{Y}|/\mathbf{L}}$.

Suppose $u \sim U(0, 1)$, and let $u = F(\mathbf{Y})$; the position of the particle relative to the attractor $\mathbf{P}$ can be obtained as $\mathbf{Y} = \pm(\mathbf{L}/2)\ln(1/u)$.

And the absolute position of the particle in the optimal space is $\mathbf{x} = \mathbf{P} + \mathbf{Y}$. Therefore, the position update formula of particle $\mathbf{x}_k$ can be obtained as follows:

$$\mathbf{x}_k(t+1) = \mathbf{P}_k(t) \pm \frac{\mathbf{L}_k(t)}{2} \ln\left(\frac{1}{u_k(t)}\right). \tag{4}$$

In (4), t is the current generation, $u_k(t) \sim U(0, 1)$. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−". $\mathbf{P}_k(t)$ and $\mathbf{L}_k(t)$ are as follows:

$$\mathbf{P}_k(t) = \varphi_k(t)\,\mathbf{pbest}_k(t) + (1 - \varphi_k(t))\,\mathbf{gbest}(t), \tag{5}$$

$$\mathbf{L}_k(t) = 2\alpha \left|\mathbf{P}_k(t) - \mathbf{x}_k(t)\right|. \tag{6}$$

In (5), $\varphi_k(t) \sim U(0, 1)$, $\mathbf{pbest}_k(t)$ is the individual best position of particle $\mathbf{x}_k$, and $\mathbf{gbest}(t)$ is the global best position. In (6), α is the expansion-constriction factor.

2.3. QPSO with Double-Well Algorithm. Based on the double-δ potential well quantum model, a QPSO with double-well algorithm is proposed by Xu et al. [14] to resolve the problem of population diversity decline due to the fast convergence of QPSO algorithm. In this algorithm, the motion of particles is simulated as in a double-well space. The double-well based position update formula of particle $\mathbf{x}_k$ can be defined as follows:

$$\begin{aligned} \mathbf{x}_k(t+1) = {} & \frac{1}{2}\left[(1 + \beta_k(t))\,\mathbf{P}_k^1(t) + (1 - \beta_k(t))\,\mathbf{P}_k^2(t)\right] \\ & \pm \frac{\mathbf{L}_k(t)}{2} \ln\left(\frac{1}{u_k(t)}\right). \end{aligned} \tag{7}$$

In (7), $\beta_k(t)$ is the distance coefficient between two δ potential wells, and $\beta_k(t)$ is a random number uniformly distributing over $[0, 1]$. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−". The two attractors $\mathbf{P}_k^1(t)$ and $\mathbf{P}_k^2(t)$ are defined as (8), and the characteristic length $\mathbf{L}_k(t)$ is defined as (9).

$$\mathbf{P}_k^1(t) = \beta_k^1(t)\,\mathbf{pbest}_k(t) + \left(1-\beta_k^1(t)\right)\mathbf{gbest}_k^1(t),$$
$$\mathbf{P}_k^2(t) = \beta_k^2(t)\,\mathbf{pbest}_k(t) + \left(1-\beta_k^2(t)\right)\mathbf{gbest}_k^2(t) \tag{8}$$

$$\mathbf{L}_k(t) = \alpha\left|\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t) - 2\mathbf{x}_k(t)\right|. \tag{9}$$

In (8), $\beta_k^1(t) \sim U(0,1)$, $\beta_k^2(t) \sim U(0,1)$, $\mathbf{pbest}_k(t)$ is the individual best position, and $\mathbf{gbest}_k^1(t)$ and $\mathbf{gbest}_k^2(t)$ are two guider particles.

3. MOQPSO-D/S Algorithm

Given the superior performance, the QPSO algorithm is introduced to solve the multiobjective problems. And the multiobjective quantum-behaved particle swarm optimization algorithm (MOQPSO) [14–18] is put forward. However, the requirement for solving a multiobjective optimization problem is different from solving a single objective optimization problem. It requires that the diversity of solutions should be good, the distribution of solutions should be uniform, and the distance between solution and the true Pareto front should be as close as possible. The fast convergence performance of QPSO would lead to the premature convergence of MOQPSO. And the global convergence to an optimal solution of QPSO would go against the diversity and distribution of solutions obtained by MOQPSO. In order to improve the performance of MOQPSO algorithm to solve the multiobjective programming problem, a multiobjective quantum-behaved particle swarm optimization with double/single-well (MOQPSO-D/S) algorithm was proposed by combining the QPSO algorithm and the QPSO with double-well algorithm. In this algorithm, a double/single-well based position update method is adopted to balance the relationship of solutions' diversity, convergence precision, and convergence rate. A hybrid random mutation method is also adopted to avoid premature convergence and improve the convergence precision. A two-stage based guider particles selection method is applied to this algorithm to improve the solution distribution uniformity and convergence rate.

3.1. Double/Single-Well Based Position Update Method. The so-called double/single-well based position update method includes two aspects. On one hand, in order to improve solutions' diversity and avoid premature convergence, the particle position is updated based on QPSO with double-well algorithm in the early optimization stage. On the other hand, in order to improve convergence precision and convergence rate, the particle position is updated based on QPSO algorithm in the final optimization stage. The double/single-well based position update formula is as follows:

$$\mathbf{x}_k(t+1) = \begin{cases} \dfrac{1}{2}\left[\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t)\right] \pm \dfrac{\mathbf{L}_k(t)}{2}\ln\left(\dfrac{1}{u_k(t)}\right), & t \le \dfrac{t_{\max}}{2} \\ \mathbf{P}_k^1(t) \pm \dfrac{\mathbf{L}_k(t)}{2}\ln\left(\dfrac{1}{u_k(t)}\right), & t > \dfrac{t_{\max}}{2}. \end{cases} \tag{10}$$

In (10), $t_{\max}$ is the maximum generation. If $u_k(t) > 0.5$, "±" would be chosen as "+", or it would be chosen as "−".

The characteristic length $\mathbf{L}_k(t)$ is defined as in the following equation.

$$\mathbf{L}_k(t) = \begin{cases} \alpha\left|\left(1+\beta_k(t)\right)\mathbf{P}_k^1(t) + \left(1-\beta_k(t)\right)\mathbf{P}_k^2(t) - 2\mathbf{x}_k(t)\right|, & t \le \dfrac{t_{\max}}{2} \\ 2\alpha\left|\mathbf{P}_k^1(t) - \mathbf{x}_k(t)\right|, & t > \dfrac{t_{\max}}{2}. \end{cases} \tag{11}$$

3.2. Hybrid Random Mutation Method. In order to avoid premature convergence and improve convergence precision, a hybrid random mutation method is proposed. The so-called hybrid random mutation is to conduct uniform random mutation in the early optimization stage of the algorithm and to conduct Gaussian random mutation in the final optimization stage. Particles can traverse on the optimal space with equal probability completely by conducting uniform random mutation, which can improve the diversity of solution and avoid premature convergence. Particles can move around the original position by conducting Gaussian mutation so as to avoid lowering optimization efficiency caused by the too large mutation range in the final optimization stage and improve convergence precision. The detail of hybrid random mutation is as follows:

$$\mathbf{x}_k(t) = \begin{cases} \mathbf{x}_k(t), & \text{rand} > p_m \\ \operatorname{Rand}\left(\mathbf{x}_{\min}, \mathbf{x}_{\max}\right), & \text{rand} \le p_m \cap t \le \dfrac{t_{\max}}{2} \\ \operatorname{Gaussian}\left(\mathbf{x}_k(t), \sigma\right), & \text{rand} \le p_m \cap t > \dfrac{t_{\max}}{2}. \end{cases} \tag{12}$$

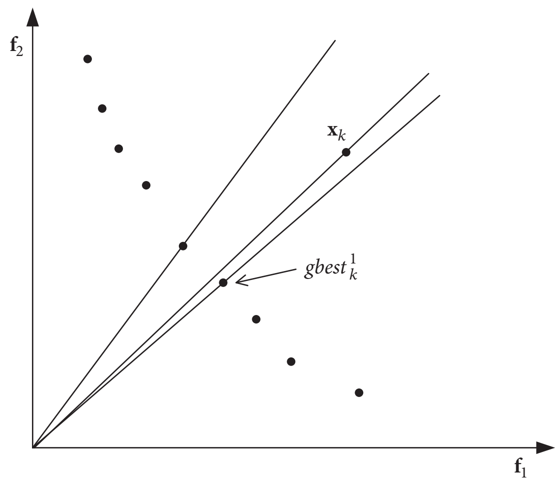

Figure 1: Sigma method based guider particle selection.

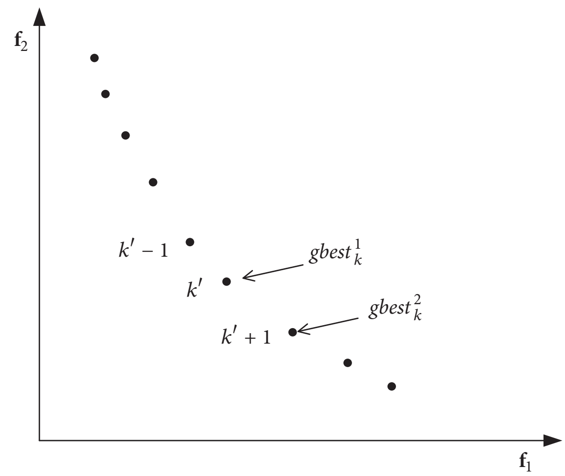

Figure 2: The $\mathbf{gbest}_k^2$ selection.

In (12), rand is a random number uniformly distributing on $[0,1]$, $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ is optimization range, Rand(•) is the uniform random mutation operator, and Gaussian(•) is the Gaussian random mutation operator. p_m is the mutation probability, which is defined as follows:

$$p_m = \begin{cases} 0.2, & t \leq \dfrac{t_{\max}}{4} \\ -\dfrac{4}{15}\dfrac{t}{t_{\max}} + \dfrac{4}{15}, & t > \dfrac{t_{\max}}{4}. \end{cases} \quad (13)$$

3.3. Two-Stage Based Guider Particles Selection Method. In MOQPSO-D/S algorithm, the guider particles for each particle should be selected during iteration. The selection of guider particles from the external archive, with application of the roulette method [17], is not conducive to the distribution uniformity of solution, due to its strong randomicity. In order to address the issue above, a two-stage based guider particles selection method is proposed. The so-called two-stage based guider particles selection method includes the following two steps. Firstly, particles in the external archive should be sorted in descending order according to their crowding distance, and the 10% of particles with the biggest crowding distance is preliminary selected. Secondly, the particle with the closest sigma value to particle $\mathbf{x}_k$ is selected as the guider particle $\mathbf{gbest}_k^1$ for $\mathbf{x}_k$. The calculation method of the crowding distance for each particle in external archive can be referred to literature [19]. The calculation method of sigma value [20] for each particle is shown as in (14), and the detail of guider particle selection by adopting sigma value method is shown in Figure 1. When the particle position is updated based on QPSO with double-well, the second guider particle should be selected. The neighbor particle, with the larger distance to $\mathbf{gbest}_k^1$, will be selected as the second guider particle $\mathbf{gbest}_k^2$. The detail of this method for $\mathbf{gbest}_k^2$ selection is shown in Figure 2.

$$\operatorname{sigma}(\mathbf{x}) = \frac{\left(f_1^2(\mathbf{x}) - f_2^2(\mathbf{x}), \ldots, f_o^2(\mathbf{x}) - f_o^2(\mathbf{x})\right)}{\sum_{l=1}^{o} f_l^2(\mathbf{x})}. \quad (14)$$

3.4. Algorithm's Complexity. Suppose that the number of objective functions in the multiobjective problem is o; the particle population is popsize; the size of external archive is N_{archive}. When operating one iteration, the complexity of MOQPSO-D/S for solving the multiobjective problem includes the following parts:

(1) The complexity of calculating the fitness of all particles equals $O(o \times \text{popsize})$.

(2) The complexity of calculating the individual best positions of all particles equals $O(o \times \text{popsize})$.

(3) After updating the positions of all particles, the complexity of comparing all particles in the population with the particles in the external archive equals $O(o \times \text{popsize} \times N_{\text{archive}})$.

(4) For updating the external archive, the complexity of conducting the crowding distance sorting equals $O(o \times (\text{popsize} + N_{\text{archive}})^2)$.

(5) After updating the external archive, the complexity of calculating the sigma values of all particles in the external archive equals $O(N_{\text{archive}})$.

Thus, when operating one iteration, the complexity of MOQPSO-D/S for solving the multiobjective problem equals $O(o \times (\text{popsize} + N_{\text{archive}})^2)$, which equals the complexity of MOQPSO-CD (the multiobjective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting) [17] and is acceptable.

4. Modeling WTA Problem Based on Fuzzy Multiobjective Programming

WTA in air and missile defense is to decide the problems of which fire unit should be chosen to intercept air targets, which air target will be intercepted by the chosen fire unit, and how many interceptors should be launched to intercept the chosen air target by the chosen fire unit. The goal of WTA decision-making is to obtain maximum interception efficiency and minimum interception consumption.

4.1. Model Assumptions and Parameter Settings. Suppose that in air and missile defense operation, m fire units $W = \{W_1, W_2, \ldots, W_m\}$ need intercept n incoming targets $T = \{T_1, T_2, \ldots, T_n\}$, where $T_1 \sim T_a$ belong to Type A (aerodynamic target), the remaining targets belong to Type B (ballistic target). Assuming that N_{A} or more interceptors are needed to intercept a target belonging to Type A and N_{B} or more interceptors are needed to intercept a target belonging to Type B, different types of targets need to be intercepted by different interceptors; for example, a target belonging to Type A can only be intercepted by Type A interceptors. The fire unit W_i ($i = 1, 2, \ldots, m$) can intercept different types of targets but can only intercept a type of targets simultaneously. Due to the sensor error and other issues, the targets threat value is uncertain. The threat value of T_j ($j = 1, 2, \ldots, n$) will be characterized with application of triangular fuzzy number $\widetilde{w}_j = (w_j^p, w_j^m, w_j^o)$, where w_j^p, w_j^m, and w_j^o denote the most pessimistic, the most likely, and the most optimistic threat value, respectively. W_i ($i = 1, 2, \ldots, m$) launches y_{ij} interceptors to intercept T_j ($j = 1, 2, \ldots, n$), and the single-shot kill probability is D_{ij}.

The correlative parameters are defined as follows:

m_i^{A} denotes the number of Type A targets that W_i can intercept simultaneously,

$\widehat{m}_i^{\mathrm{A}}$ denotes the number of Type A interceptors that W_i can launch simultaneously,

N_i^{A} denotes the number of Type A interceptors that W_i stores,

C_i^{A} denotes the value of a Type A interceptor that W_i stores,

m_i^{B} denotes the number of Type B targets that W_i can intercept simultaneously,

$\widehat{m}_i^{\mathrm{B}}$ denotes the number of Type B interceptors that W_i can launch simultaneously,

N_i^{B} denotes the number of Type B interceptors that W_i stores,

C_i^{B} denotes the value of a Type B interceptor that W_i stores.

4.2. Modeling Based on Fuzzy Multiobjective Programming. Taking the maximizing interception efficiency and the minimizing interception consumption as goals, a multiobjective optimization based WTA model is established as follows.

$$\max \quad f_1 = \sum_{j=1}^{n} \widetilde{w}_j \left[1 - \prod_{i=1}^{m} \left(1 - D_{ij}\right)^{y_{ij}}\right], \tag{15}$$

$$\min \quad f_2 = \sum_{i=1}^{m} \left[\sum_{j=1}^{a} C_i^A y_{ij} + \sum_{j=a+1}^{n} C_i^B y_{ij}\right]. \tag{16}$$

The model needs to meet the resource constraints and the firepower constraint. For the fire unit W_i ($i = 1, 2, \ldots, m$), the resource constraints are needed to be met as follows.

(1) Interception Constraint. The fire unit W_i can only intercept a type of target simultaneously; thus the model should meet

$$\delta\left(\sum_{j=1}^{a} y_{ij}\right) + \delta\left(\sum_{j=a+1}^{n} y_{ij}\right) \leq 1. \tag{17}$$

In (17), $\delta(\xi) = \begin{cases} 1, & \xi > 0 \\ 0, & \xi \leq 0 \end{cases}$.

(2) Interception Capability Constraint. The fire unit W_i can intercept less than m_i^{A} Type A targets or m_i^{B} Type B targets simultaneously; thus the model should meet

$$\begin{aligned} \sum_{j=1}^{a} \delta\left(y_{ij}\right) &\leq m_i^{\mathrm{A}} \\ \sum_{j=a+1}^{n} \delta\left(y_{ij}\right) &\leq m_i^{\mathrm{B}}. \end{aligned} \tag{18}$$

(3) Number Constraint of Interceptors. The fire unit W_i can launch less than $\widehat{m}_i^{\mathrm{A}}$ and N_i^{A} to intercept Type A targets or launch less than $\widehat{m}_i^{\mathrm{B}}$ and N_i^{B} to intercept Type B targets; thus the model should meet

$$\begin{aligned} \sum_{j=1}^{a} y_{ij} &\leq \min\left(\widehat{m}_i^{\mathrm{A}}, N_i^{\mathrm{A}}\right) \\ \sum_{j=a+1}^{n} y_{ij} &\leq \min\left(\widehat{m}_i^{\mathrm{B}}, N_i^{\mathrm{B}}\right). \end{aligned} \tag{19}$$

For intercepting the incoming target T_j ($j = 1, 2, \ldots, n$), the firepower constraint is needed to be met as follows.

N_{A} or more Type A interceptors are needed to intercept a Type A target, and N_{B} or more Type B interceptors are needed to intercept a Type B target; thus the model should meet

$$\begin{aligned} \sum_{i=1}^{m} y_{ij} &\geq N_{\mathrm{A}}, \quad j = 1, 2, \ldots, a \\ \sum_{i=1}^{m} y_{ij} &\geq N_{\mathrm{B}}, \quad j = a+1, a+2, \ldots, n. \end{aligned} \tag{20}$$

4.3. Clarifying Model Equivalently. This WTA model cannot be resolved by adopting multiobjective optimization algorithm, for its objective function containing fuzzy parameter shown as (15). Therefore, it is necessary to clarify the fuzzy multiobjective programming based WTA model to be a certain model equivalently. The necessity degree principle of uncertainty theory [21] is adopted to turn the objective function with fuzzy parameter into a certain objective function. The detailed process is as follows.

Firstly, the objective function shown as (15) is turned into a fuzzy chance constrained programming based model

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & \operatorname{Nec}\left\{\sum_{j=1}^{n}\widetilde{\omega}_j\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}} \le c\right\} \ge \theta. \end{aligned} \tag{21}$$

In (21), θ denotes the necessity degree that the whole rest incoming targets' threat value is equal to or less than c.

Secondly, the fuzzy chance constrained programming based model shown as in (21) is turned into certain form with application of the necessity degree principle of uncertainty theory. Because $\operatorname{Nec}\{\sum_{j=1}^{n}\widetilde{\omega}_j\prod_{i=1}^{m}(1-D_{ij})^{y_{ij}} \le c\} \ge \theta$ can be turned into $\sum_{j=1}^{n}[w_j^m+\theta(w_j^o-w_j^m)]\prod_{i=1}^{m}(1-D_{ij})^{y_{ij}} \le c$ equivalently, (17) can be turned into

$$\begin{aligned} \min \quad & c \\ \text{s.t.} \quad & \sum_{j=1}^{n}\left[w_j^m+\theta\left(w_j^o-w_j^m\right)\right]\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}} \le c. \end{aligned} \tag{22}$$

Finally, the certain objective function is obtained as follows:

$$\min \quad f_1' = \sum_{j=1}^{n}\left[w_j^m+\theta\left(w_j^o-w_j^m\right)\right]\prod_{i=1}^{m}\left(1-D_{ij}\right)^{y_{ij}}. \tag{23}$$

In this way, the fuzzy multiobjective programming based WTA model is turned into a certain multiobjective optimization model. The model clarified could be solved with application of multiobjective optimization algorithm. The WTA model constructed in this paper will be solved by adopting MOQPSO-D/S algorithm in the next section.

5. Solving WTA Model Based on MOQPSO-D/S

5.1. Particle Coding. The MOQPSO-D/S algorithm cannot solve the discrete problem directly. Therefore, it is necessary to code the position of particles when the WTA model is resolved by adopting MOQPSO-D/S algorithm. The position of particles will be coded with application of matrix encoding. $\mathbf{y}$ denotes the position of a particle, and it can be coded as follows:

$$\mathbf{y} = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix}. \tag{24}$$

In (24), y_{ij} denotes the number of interceptors that W_i launches to intercept T_j.

5.2. Illegal Particle Code Adjusting. When initializing and updating the particle position, the particle position may not satisfy the constraints of the WTA model; namely, the infeasible solution comes out. Therefore, it is necessary to check the particle coding legitimacy and to adjust the illegal particle code. The process of particle code checking and illegal code adjusting is shown in Figure 3.

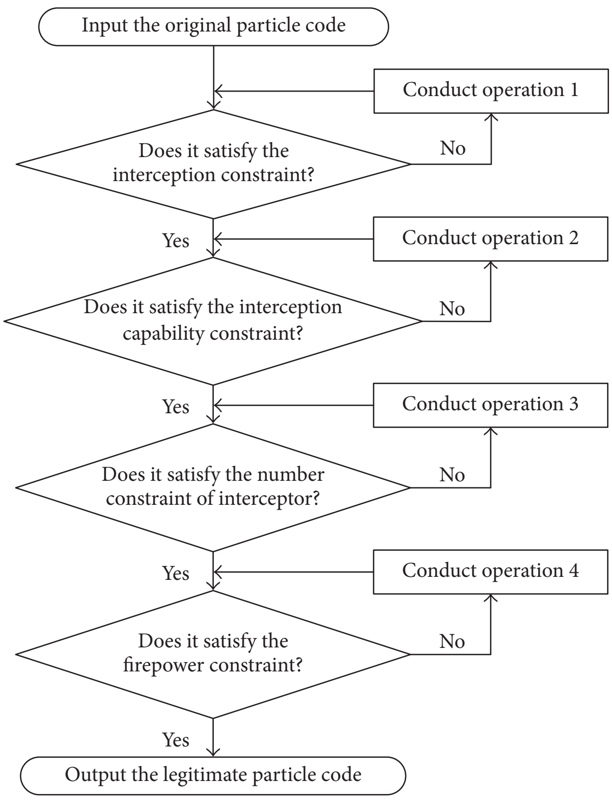

Figure 3: The process of particle code checking and illegal code adjusting.

For example, when a particle code does not meet the interception capability constraint for fire unit W_i ($i = 1, 2, \ldots, m$), Operation 2 should be conducted as follows:

Firstly, judge whether $\sum_{j=1}^{a} y_{ij} = 0$ or $\sum_{j=a+1}^{n} y_{ij} = 0$.

Case 1. One has

$$\sum_{j=1}^{a} y_{ij} = 0 \tag{25}$$

C1S1 (Case 1 Step 1). Let count_A $= \sum_{j=1}^{a}\delta(y_{ij})$; if $\delta(y_{ij}) = 1$, then store the subscript j into the set Ω_A.

C1S2. Let $N_{\text{exc\_A}} = \text{count\_A-}m_i^{A}$.

C1S3. Select randomly $N_{\text{exc\_A}}$ subscript from the set Ω_A, and let the corresponding $y_{ij} = 0$.

Case 2. One has

$$\sum_{j=a+1}^{n} y_{ij} = 0 \tag{26}$$

C2S2. Let count_B $= \sum_{j=a+1}^{n} \delta(y_{ij})$; if $\delta(y_{ij}) = 1$, then store the subscript j into the set Ω_{B}.

C1S2. Let $N_{\mathrm{exc\_B}}$ = count_B-m_i^{B}.

C1S3. Select randomly $N_{\mathrm{exc\_B}}$ subscript from the set Ω_{B}, and let the corresponding $y_{ij} = 0$.

Finally, output the particle code that meets the interception capability constraint for fire unit W_i $(i = 1, 2, \ldots, m)$.

Due to limitations on space, Operation 1, Operation 3, and Operation 4 will not be introduced here.

5.3. Solving Steps. After the particles are encoded and the illegal particles are adjusted, the MOQPSO-D/S algorithm can be adopted to solve the fuzzy multiobjective programming based WTA model clarified equivalently. The detailed steps are as follows.

Step 1. Set the algorithm parameters, and initialize the particle swarm randomly.

Step 2. Check out whether every particle code meets the constraints. If a particle code does not meet the constraints, then adjust it. Otherwise, turn to Step 3.

Step 3. Calculate the fitness of every particle $f(\mathbf{y}) = (f_1^f(\mathbf{y}), f_2(\mathbf{y}))$ according to (16) and (23); let $\mathbf{pbest} = \mathbf{y}$, $f(\mathbf{pbest}) = f(\mathbf{y})$, where **pbest** denotes the individual best position of $\mathbf{y}$.

Step 4. Initialize the external storage file ARCHIVE. Calculate the dominated relationship between particles according to their fitness, and store the nondominated particles into ARCHIVE, then calculate the crowding distance and sigma value of particles in ARCHIVE.

Step 5. Initialize iteration, and let $t = 1$.

Step 6. Select guider particles $\mathbf{gbest}^1$ and $\mathbf{gbest}^2$ with application of two-stage method. If $t > t_{\max}/2$, the guider particle $\mathbf{gbest}^2$ would not be selected. Update the position of every particle according to (10).

Step 7. Judge whether the mutation condition is satisfied or not. If it is satisfied, conduct mutation for the particle by adopting the hybrid random mutation method. Otherwise, turn to Step 8.

Step 8. Check out whether every particle code meets the constraints. If a particle code does not meet the constraints, then adjust it. Otherwise, turn to Step 9.

Step 9. Calculate the fitness of every particle $f(\mathbf{y}(t)) = (f_1^f(\mathbf{y}(t)), f_2(\mathbf{y}(t)))$, and calculate the dominated relationship between the current position and the individual best position of every particle. If the current position dominates the individual best position of the particle $\mathbf{y}_k$, then let $\mathbf{pbest}_k = \mathbf{y}_k(t)$, $f(\mathbf{pbest}_k) = f(\mathbf{y}_k(t))$. If the current position and the individual best position do not dominate each other, then conduct the operation $\mathbf{pbest}_k = \mathbf{y}_k(t)$, $f(\mathbf{pbest}_k) = f(\mathbf{y}_k(t))$ with the probability of 0.5. Otherwise, turn to Step 10.

Step 10. Calculate the dominated relationship between every particle and particles in ARCHIVE. If the particle $\mathbf{y}_k$ dominates the particle $\mathbf{y}_g$ in ARCHIVE, then store the particle $\mathbf{y}_k$ into ARCHIVE, delete the particle $\mathbf{y}_g$, and turn to Step 11. If the particle $\mathbf{y}_k$ and the particle $\mathbf{y}_g$ do not dominate each other, then store the particle $\mathbf{y}_k$ into ARCHIVE, and turn to Step 11. Otherwise, turn to Step 12.

Step 11. Calculate the crowding distance of particles in ARCHIVE updated, and calculate the sigma value of the particles newly stored. Judge whether the number of particles in ARCHIVE exceeds its capacity. If it exceeds the capacity, then delete the particle with the smallest crowning distance. Otherwise, turn to Step 12.

Step 12. Let $t = t + 1$. If $t > t_{\max}$, then end the iteration, and output all the particles in ARCHIVE as the WTA alternatives for air and missile defense combat. Otherwise, turn to Step 6.

6. Simulation Experiment and Analysis

In order to verify the performance of MOQPSO-D/S algorithm for solving multiobjective optimization problems and WTA problem in air and missile defense operation, two simulation experiments are designed. In experiment 1, the convergence and solution distribution uniformity of the MOQPSO-D/S algorithm for solving multiobjective optimization problem are validated. In experiment 2, after the simulation scenarios are set, the rationality of the fuzzy multiobjective programming based WTA model and the feasibility of the MOQPSO-D/S algorithm for solving the model constructed in this paper are validated by solving the WTA model with application of MOQPSO-D/S algorithm coded. The experiments above are conducted on the computer with INTEL CORE i5-4590, 3.3 GHz CUP, and 4 G RAM after programming with MATLAB 2013a. The capability of ARCHIVE for all multiobjective optimization algorithms is 100.

6.1. Verify the Performance of MOQPSO-D/S for Solving Multiobjective Optimization Problems. The test functions ZDT1~ZDT4 [22] are chosen to verify the performance of MOQPSO-D/S algorithm for solving multiobjective optimization problems. The multiobjective quantum-behaved particle swarm optimization algorithm with double-potential well and share-learning (MOQPSO-DPS) [14], the MOQPSO-CD [17], and the multiobjective particle swarm optimization algorithm based on crowding distance sorting (MOPSO-CD) [23] are chosen as the comparative algorithms. Set the particle population to be 100, and set the iteration number to be 200. The variables number of the test functions ZDT1~ZDT3 is 30, and the variables number of the test function ZDT4 is 10. The algorithms run 30 times independently. Select the solution of one time from the 30 times of every algorithm for solving test functions ZDT1~ZDT4 randomly, and the results are shown in Figure 4.

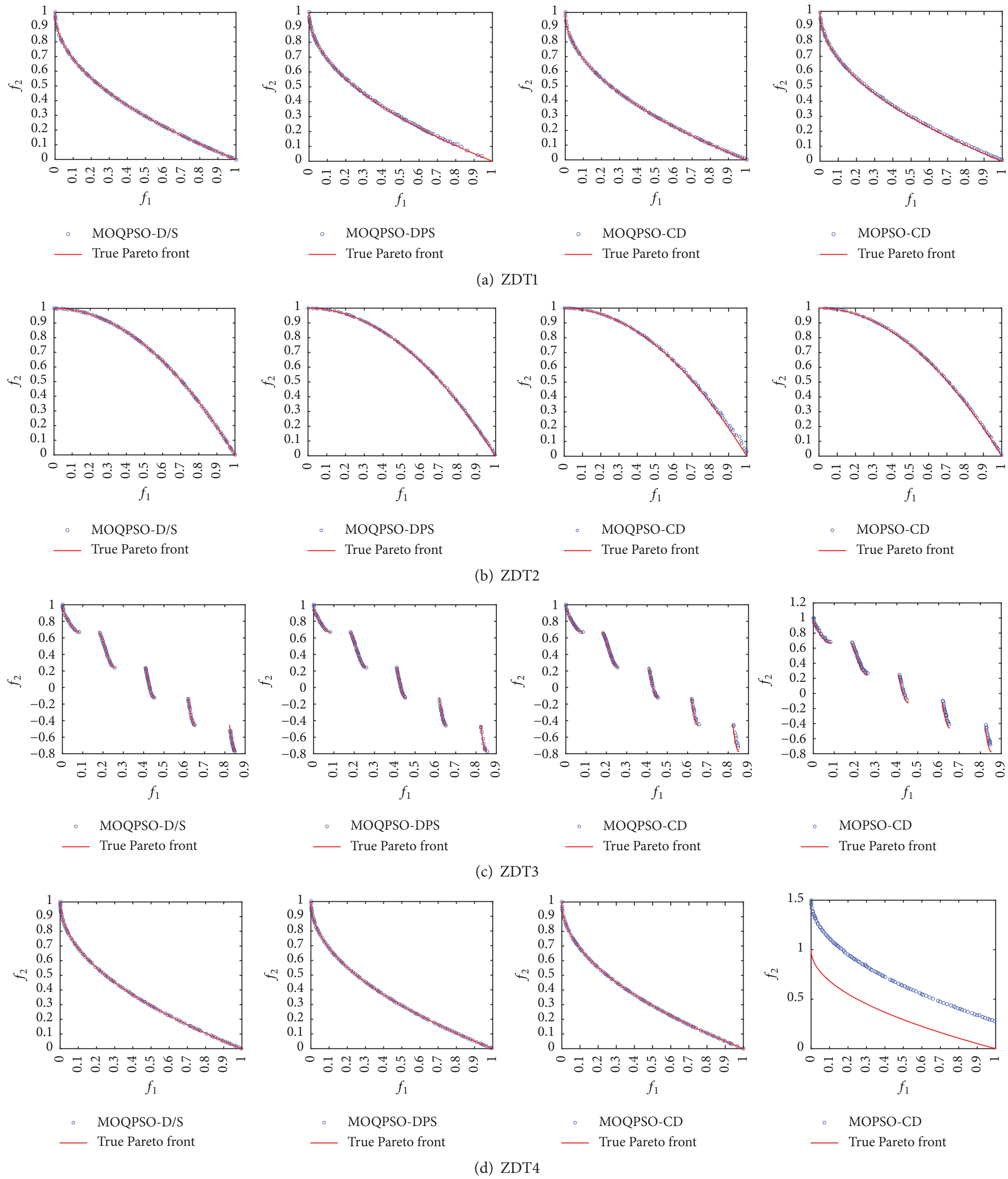

Figure 4: Pareto front obtained by solving ZDT1~ZDT4 with application of four algorithms.

Table 1: ER of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**0.0097**	**0.0087**	**0.0083**	**0.0290**
MOQPSO-DPS	0.0150	0.0127	0.0187	0.0370
MOQPSO-CD	0.0113	0.0253	0.0343	0.0337
MOPSO-CD	0.0283	0.0363	0.1143	0.2783

Table 2: SP of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**0.0043**	0.0075	**0.0089**	**0.0074**
MOQPSO-DPS	0.0117	**0.0073**	0.0114	0.0131
MOQPSO-CD	0.0071	0.0101	0.0121	0.0087
MOPSO-CD	0.0079	0.0084	0.0143	0.0241

Table 3: Mean running time of four algorithms for solving the test functions.

Algorithms	Test functions			
	ZDT1	ZDT2	ZDT3	ZDT4
MOQPSO-D/S	**5.1134**	**6.7634**	**4.0197**	**4.5163**
MOQPSO-DPS	7.2471	7.3521	4.4110	6.0217
MOQPSO-CD	8.3283	8.1126	4.7853	6.4313
MOPSO-CD	10.3590	11.8781	6.7596	8.8357

Error ration (ER) [14] and spacing metric (SP) [24] are chosen to weigh the convergence of four algorithms for solving the test functions and the distribution uniformity of the Pareto optimal solution, respectively. And, the mean running time is chosen to weigh the operating rate. The calculations of ER and SP are shown in (27) and (28). ER index represents the ratio of the Pareto solutions obtained by four algorithms in the true Pareto optimal solution set. The smaller the ER value is, the better the convergence is. The smaller the SP value is, the more uniformly the Pareto optimal solutions distribute.

$$\mathrm{ER} = \frac{\sum_{i=1}^{n_P} e_i}{n_P}. \tag{27}$$

In (27), e_i denotes the probability that a Pareto optimal solution is not in the true Pareto optimal solution set, if it is in, $e_i = 0$; or, $e_i = 1$. n_P denotes the number of elements in a Pareto optimal solution set.

$$\mathrm{SP} = \sqrt{\frac{1}{n_P - 1} \sum_{k=1}^{n_P} \left(\overline{d} - d_k\right)^2}. \tag{28}$$

In (28), d_k denotes the Hamming distance between the dot corresponding to solution $\mathbf{x}_k$ and its neighbor dot in the Pareto front. The calculation of d_k is shown as in (29). $\overline{d}$ denotes the mean distance, shown in (30).

$$d_k = \min_{r \neq k} \left(\sum_{\xi=1}^{o} \left| f_\xi\left(\mathbf{x}_k\right) - f_\xi\left(\mathbf{x}_r\right) \right| \right), \tag{29}$$

$$\overline{d} = \frac{1}{n_P} \sum_{k=1}^{n_P} d_k. \tag{30}$$

The comparative results are shown in Tables 1–3, where the ER values, SP values, and the mean running time values are the average values of four algorithms for solving the test functions 30 times.

From Table 1, we can see that the ER indexes of MOQPSO-D/S for solving the test functions are smaller than the remaining algorithms'. Combining with Figure 4, it can be concluded that the convergence of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the double/single-well based position update method and the hybrid random mutation method can improve convergence precision.

From Table 2, we can see that the SP indexes of MOQPSO-D/S for solving the test functions are smaller than the remaining algorithms', except for solving ZDT2. Combining with Figure 4, it can be concluded that the distribution uniformity of the Pareto optimal solution of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the two-stage based guider particles selection method can improve solution distribution uniformity.

Table 4: Parameters value.

	W_1	W_2	W_3	W_4	W_5	W_6
m_i^a	1	3	2	5	4	0
$\widehat{m}_i^a$	3	6	4	10	8	0
m_i^b	0	0	0	1	1	1
$\widehat{m}_i^b$	0	0	0	2	2	2
N_i^a	6	6	8	9	8	0
N_i^b	0	0	0	2	2	2
C_i^a	0.1	0.2	0.15	0.25	0.3	-
C_i^b	-	-	-	0.45	0.5	0.6

Table 5: Single-shot kill probability.

	W_1	W_2	W_3	W_4	W_5	W_6
T_1	0.45	0.32	0.54	0.47	0.75	0
T_2	0.5	0.27	0.34	0.73	0.78	0
T_3	0.28	0.62	0.58	0.56	0.57	0
T_4	0	0	0	0.63	0.55	0.7
T_5	0	0	0	0.45	0.5	0.85

From Table 3, we can see that the mean running time of MOQPSO-D/S for solving the test functions is smaller than the remaining algorithms'. It can be concluded that the operating rate of MOQPSO-D/S is better than the remaining algorithms'. And, it is proved that the two-stage based guider particles selection method can improve operating rate.

6.2. Verify the Performance of MOQPSO-D/S for Solving Uncertain WTA Problem. In order to validate the rationality of the fuzzy multiobjective programming based WTA model and to verify the performance of the MOQPSO-D/S algorithm for solving the model simultaneously, the following case for air and missile defense combat is given.

Suppose that, in air and missile defense operation, 6 fire units W_i $(i = 1, 2, \dots, 6)$ need intercept 5 incoming targets T_j $(j = 1, 2, \dots, 5)$. The number of targets that W_i $(i = 1, 2, \dots, 6)$ can intercept simultaneously, the number of interceptors that W_i can launch simultaneously, the number of interceptors that W_i stores, and the dimensionless value of an interceptor that W_i stores are listed in Table 4. The incoming targets $T_1 \sim T_3$ belong to Type A, and the remaining targets belong to Type B. The target threat values are weighed with application of triangular fuzzy number as follows: $\widetilde{\omega}_1 = (0.41, 0.43, 0.45)$, $\widetilde{\omega}_2 = (0.48, 0.51, 0.54)$, $\widetilde{\omega}_3 = (0.33, 0.35, 0.37)$, $\widetilde{\omega}_4 = (0.87, 0.89, 0.91)$, $\widetilde{\omega}_5 = (0.90, 0.92, 0.94)$. The least interceptors to intercept a Type A target and a Type B target are $N_A = 1$ and $N_B = 2$, respectively. The necessity degree θ is set to be 0.8. The single-shot kill probability of the fire unit W_i $(i = 1, 2, \dots, 6)$ that launches an interceptor to intercept the target T_j $(j = 1, 2, \dots, 5)$ is listed in Table 5. The particle encoding method and the illegal particle code adjusting method are adopted in the subsequent algorithms for solving the example.

6.2.1. Analyze the Result of MOQPSO-D/S for Solving Uncertain WTA Problem. With different population size and different iterations, the WTA case is solved with application of MOQPSO-D/S algorithm. The algorithm for solving the WTA case under every condition runs 30 times independently. The mean running time is listed in Table 6. It can be seen from Table 6 that the real-time performance of MOQPSO-D/S algorithm can meet the need of WTA in air and missile defense combat.

Table 6: Mean running time/s.

Population size	Iterations		
	50	100	200
50	0.7504	1.4830	2.9733
100	1.5179	2.9313	5.9883
200	3.0576	6.0737	11.8876

With the population size and iterations both being 50, the solution of the WTA case solved by MOQPSO-D/S algorithm is shown in Figure 5. It can be seen from Figure 5 that the Pareto optimal solution, obtained by MOQPSO-D/S algorithm for solving the fuzzy multiobjective programming based WTA model in air and missile defense combat, distributes uniformly, where every dot refers to a WTA alternative. Therefore, it can provide a series of alternatives for commanders to make decision. The commanders could make the final decision by choosing a suitable WTA alternative from the solution with consideration of the battlefield situation and risk preference. Comparing with the single objective optimization based WTA method, which can only give a WTA alternative, this method could preferably adapt to the change of the battlefield situation and combine the tactical thought of the commander to make decision.

6.2.2. Compare with the Single Objective Optimization Based WTA Method. In order to validate the feasibility of the solution obtained by MOQPSO-D/S algorithm for solving WTA model, in comparison, the example is solved by adopting the method proposed in literature [6], taking maximizing effectiveness-cost ratio as the objective function and adopting

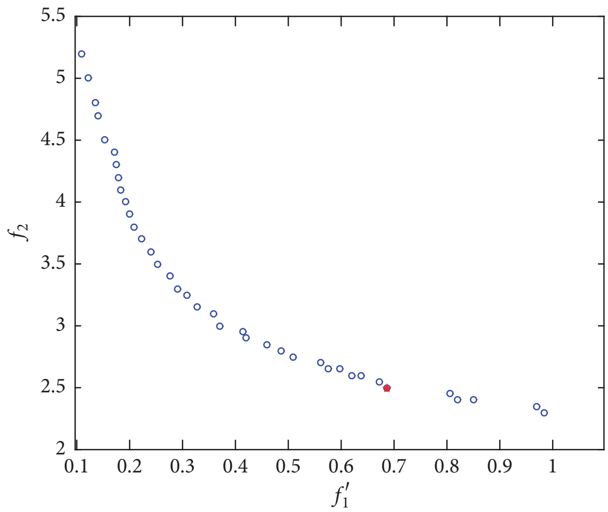

Figure 5: Solution of MOQPSO-D/S for solving the case.

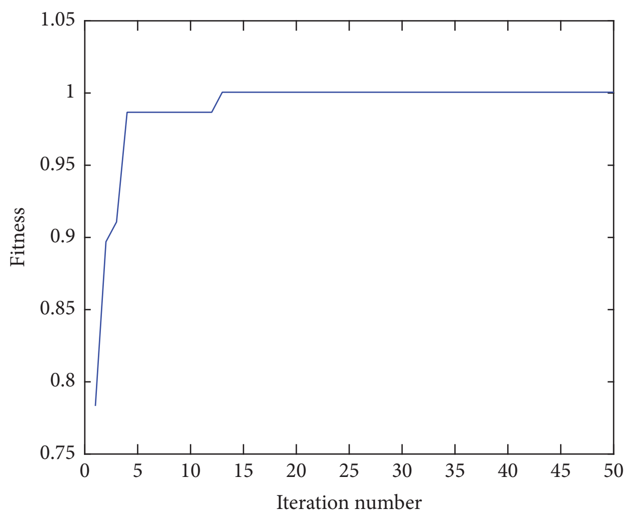

Figure 6: Best fitness over generations based on IDPSO algorithm.

IDPSO algorithm to solve the WTA model. With the population size and iterations being both 50, the single objective optimization method for solving the WTA case runs 30 times. The mean running time is 0.7193 s. The best fitness over generations based on IDPSO is shown as in Figure 6. The optimal WTA result is

$$\begin{array}{c} \\ W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \end{array} \begin{array}{c} \begin{array}{ccccc} T_1 & T_2 & T_3 & T_4 & T_5 \end{array} \\ \begin{bmatrix} 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{31}$$

The optimal result is just the WTA alternative corresponding to the dot marked by a red five-angle star shown in Figure 5. And the mean running time is a little shorter than the MOQPSO-D/S algorithm with the same population size and iterations. It is proved that MOQPSO-D/S algorithm, when solving the WTA model, would disperse computation resource but can still obtain the satisfactory solution containing the optimal result obtained by single optimization method. Meanwhile, the fuzzy multiobjective programming WTA model for air and missile defense is proved to be reasonable and adopting MOQPSO-D/S algorithm to solve the model is feasible.

Table 7: Dominated ratio among four algorithms' Pareto optimal sets.

X_2	X_1			
	A_1	A_2	A_3	A_4
A_1	0	0	0.0185	0.0370
A_2	0.0588	0	0.1176	0.1765
A_3	0.0556	0	0	0
A_4	0.0755	0.0755	0.0755	0

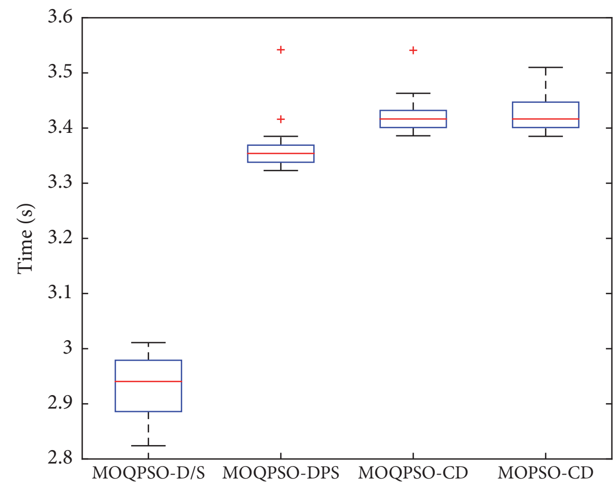

Figure 7: Statistical results for running time of four algorithms.

6.2.3. Compare with Other Multiobjective Optimization Algorithm for Solving WTA Method. In order to further verify the performance of MOQPSO-D/S algorithm for solving WTA problem, in comparison, the example is solved by adopting MOQPSO-DPS, MOQPSO-CD, and MOPSO-CD algorithm with the population size and iterations being both 100. Every algorithm for solving the WTA case runs 30 times independently. The statistical results for running time of every algorithm are shown in Figure 7. It can be seen from Figure 7 that the real-time performance of MOQPSO-D/S algorithm is obviously superior to the comparable algorithms'.

The convergence of every algorithm for solving the WTA case is weighed by dominated ratio $C(X_1, X_2)$ [25]. The calculation of $C(X_1, X_2)$ is shown as in (32). X_1 and X_2 denote the Pareto optimal solution sets obtained by two algorithms, respectively. If $C(X_1, X_2)$ is bigger than $C(X_2, X_1)$, the convergence of the algorithm corresponding to X_1 is superior to the algorithm corresponding to X_2. The convergence comparison among four algorithms is listed in Table 7. A_1, A_2, A_3, and A_4 listed in Table 7 denote the

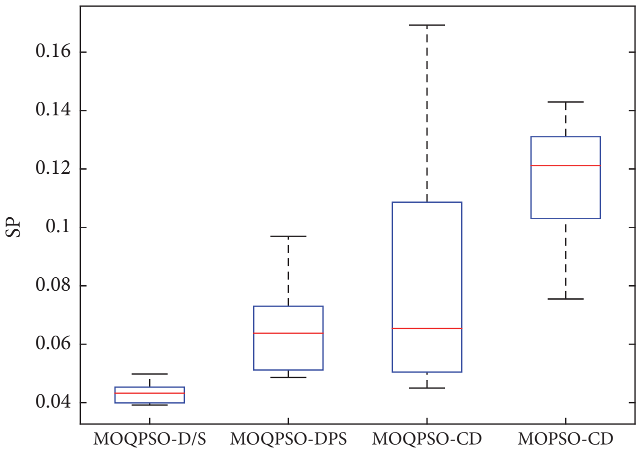

Figure 8: Statistical results for SP value of four algorithms.

Pareto optimal solution sets obtained by MOQPSO-D/S, MOQPSO-DPS, MOQPSO-CD, and MOPSO-CD algorithm, respectively. It can be concluded from Table 7 that the convergence of MOQPSO-D/S algorithm for solving the WTA case is the best of four algorithms.

$$C(X_1, X_2) = \frac{\operatorname{numel}(\{\mathbf{x}_2 \in X_2 \mid \exists \mathbf{x}_1 \in X_1 : \mathbf{x}_1 \prec \mathbf{x}_2\})}{\operatorname{numel}(X_2)}. \quad (32)$$

In (32), the function numel(•) is used to count the number of elements in a set.

The distribution uniformity of the Pareto optimal solution set obtained by every algorithm for solving the WTA case is weighed by SP. The statistical results for SP value of the Pareto optimal solution set obtained by every algorithm are shown in Figure 8. It can be concluded from Figure 8 that the Pareto optimal solutions obtained by MOQPSO-D/S algorithm for solving the WTA case distribute the most uniformly, and its distribution uniformity is the most stable.

7. Conclusions

The goal of this article is to find a method to aid decision-making of WTA in air and missile defense combat under uncertainty. A fuzzy multiobjective programming based WTA model has been constructed. And multiobjective optimal algorithm, called MOQPSO-D/S, has been put forward to solve the fuzzy multiobjective programming based WTA model. The simulation results shows that the performance of MOQPSO-D/S algorithm is superior, the WTA model constructed in this paper is rational and the MOQPSO-D/S algorithm can solve the WTA model efficiently.

In future research, we may study WTA problem under uncertainty based on other uncertain methods, for example, the Taguchi method and the Bayesian method. Yet, we may study multiobjective optimization algorithms, for example, the game theory based methodologies, to solve the multiobjective programming based WTA model.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by National Natural Science Foundation of China (Grant no. 71771216 and Grant no. 71701209).

References

[1] J. M. Teague, K. R. Dorner, and W. B. Hartman, *Back to the future: integrated air and missile defense in the pacific*, Air University, Air Force Research Institute USA, Maxwell, USA, 2015.

[2] M. Ni, Z. Yu, F. Ma, and X. Wu, "A lagrange relaxation method for solving weapon-target assignment problem," *Mathematical Problems in Engineering*, vol. 2011, Article ID 873292, 10 pages, 2011.

[3] T. Q. Chang, J. W. Chen, N. Hao et al., "Terminating control of ant colony algorithm for armored unit," *System Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.

[4] R. Wang H and C. Wang, "Variable value control technology of genetic algorithm for wta of ground target attacking," *Acta Armamentraii*, vol. 37, no. 10, pp. 1889–1895, 2016.

[5] H. Liang and F. Kang, "Adaptive chaos parallel clonal selection algorithm for objective optimization in WTA application," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 6, pp. 3459–3465, 2016.

[6] C. L. Fan, Q. H. Xing, M. F. Zheng et al., "Weapon-target allocation optimization algorithm based on IDPSO," *System Engineering and Electronics*, vol. 37, no. 2, pp. 336–342, 2015.

[7] X. Liu, Z. Liu, W. S. Hou et al., "Improved MOPSO algorithm for multi-objective programming model of weapon-target assignment," *Journal of Systems Engineering and Electronics*, vol. 35, no. 2, pp. 326–330, 2013.

[8] Y. Zhang, R. N. Yang, and J. L. Zuo, "Improved decomposition-based evolutionary algorithm for multi-objective optimization model of dynamic weapon-target assignment," *Acta Armamentraii*, vol. 36, no. 8, pp. 1533–1540, 2015.

[9] Y. Zhang, R. N. Yang, J. L. Zuo et al., "Improved MOEA/D for dynamic weapon-target assignment problem," *Journal of Harbin Institute of Technology*, vol. 22, no. 6, pp. 121–128, 2015.

[10] J. Sun, B. Feng, and W. B. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 325–331, IEEE, Portland, OR, USA, 2004.

[11] J. Sun, W. B. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 111–116, IEEE, Singapore, Singapore, 2004.

[12] Y. J. Won, K. H. Lee, and J. H. Lee, "Performance improvement of space borne SAR using antenna pattern synthesis based on quantum-behaved particle swarm optimization," *International Journal of Antennas and Propagation*, pp. 1–12, 2017.

[13] M. Xu, L. Zhang, B. Du et al., "A mutation operator accelerated quantum-behaved particle swarm optimization algorithm for hyperspectral endmember extraction," *Remote Sensing*, vol. 9, no. 3, p. 197, 2017.

[14] S.-H. Xu, X.-D. Mu, D. Chai, and P. Zhao, "Multi-objective quantum-behaved particle swarm optimization algorithm with

double-potential well and share-learning," *Optik - International Journal for Light and Electron Optics*, vol. 127, no. 12, pp. 4921–4927, 2016.

[15] Y. Li, Y. Wang, and L. Jiao, "MOQPSO: A new quantum-behaved particle swarm optimization for nearest neighborhood classification," in *Proceedings of the 2016 IEEE Region 10 Conference, TENCON 2016*, pp. 3165–3168, Singapore, November 2016.

[16] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. Narayana Naik, and S. Gopalakrishnan, "Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures," *Expert Systems with Applications*, vol. 36, no. 8, pp. 11312–11322, 2009.

[17] Z. Shi and Q. W. Chen, "Multi-objective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting," *Control and Decision*, vol. 26, no. 4, pp. 540–547, 2011.

[18] A. B. Heyam, *A quantum behaved particle swarm approach to multi-objective optimization*, University of Birmingham, Birmingham, England, 2015.

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

[20] S. Mostaghim and J. Teich, "Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO)," in *Proceedings of the IEEE Swarm Intelligence Symposium*, pp. 26–33, IEEE, Indianapolis, IN, USA, April 2003.

[21] B. D. Liu, "Uncertainty Theory," (Fifth Edition), Uncertainty Theory Laboratory, 2016, http://orsc.edu.cn/liu/ut.pdf.

[22] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[23] C. R. Raquel and P. C. Naval Jr., "An effective use of crowding distance in multiobjective particle swarm optimization," in *Proceedings of the 7th Annual conference on Genetic and Evolutionary Computation*, pp. 257–264, ACM, New York, NY, USA, June 2005.

[24] J. R. Schott, *Fault tolerant design using single and multicriteria genetic algorithm optimization*, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[25] E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization Methods and Applications*, Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, 1999.