International Journal of Innovative Computing, Information and Control
Volume 11, Number 6, December 2015
ICIC International ©2015 ISSN 1349-4198
pp. 1835–1853

A CONSTRAINED MULTI-OBJECTIVE PARTICLE SWARM OPTIMIZATION ALGORITHM BASED ON ADAPTIVE PENALTY AND NORMALIZED NON-DOMINATED SORTING

Weidong Zhang[1], Xianlin Huang[1], Xiao-Zhi Gao[2] and Hang Yin[1]

[1]Centre of Control Theory and Guidance Technology
Harbin Institute of Technology
No. 92, West Dazhi Street, Nangang District, Harbin 150001, P. R. China
hifinemoment@163.com; { xlinhuang; yh }@hit.edu.cn

[2]Department of Automation and Systems Technology
School of Electrical Engineering
Aalto University
02150 Espoo, Helsinki, Finland
xiao-zhi.gao@aalto.fi

Received March 2015; revised July 2015

Abstract. *In order to deal with constrained multi-objective optimization problems (CMOPs), a novel constrained multi-objective particle swarm optimization (CMOPSO) algorithm is proposed based on an adaptive penalty technique and a normalized non-dominated sorting technique. The former technique is utilized to optimize constrained individuals in each generation to obtain new objective functions, while the latter technique ranks individuals along with the new objective functions obtained from the adaptive penalty technique. Additionally, the external archive maintenance has been improved by external population size decrease, and selection of individuals with better ranks which are operated by Pareto constrained-dominance. Based on the concept of crowding distance, the global best solution is obtained and the individuals of the next generation are provided by the basic PSO algorithm. The results of the simulation tests indicate precise convergence and diverse distribution of the non-dominant solutions on true Pareto front, which demonstrates that the proposed algorithm possesses outstanding performance metrics for generational distance and spacing. Finally, the trajectory optimization problem for hypersonic reentry glide vehicles (HRGVs) applied further verifies the effectiveness and efficiency of the proposed CMOPSO algorithm, which shows a good application prospect of the proposed algorithm as well.*
Keywords: CMOPs, CMOPSO, Adaptive penalty, Normalized non-dominated sorting, External archive, Pareto constrained-dominance, Crowding distance

1. **Introduction.** A large amount of real-life and engineering problems belong to multi-objective optimization problems (MOPs), which have multiple conflicting performance indexes or objectives to be optimized simultaneously to achieve a tradeoff, such as aerospace systems, electrical systems, biological sciences and data mining [1]. Thus, the problem of multi-objective optimization arises. Furthermore, if decision variables need to meet certain constraints, new optimization challenge appears, thus forming the so called constrained multi-objective optimization problems (CMOPs). In fact, the research of these MOPs, including CMOPs, has become a common concern in academics and engineering applications, and possesses an important practical significance.

For solving CMOPs, traditional gradient-based search methods, such as the projected gradient method and quadratic programming method [2], are difficult to extend to the multi-objective case because their basic design precludes the consideration of multiple solutions. What is more, the requirements of sufficient gradient information and appropriate initialization make them powerless for cases with non-differentiable, discontinuous, and implicit functions.

As population-based metaheuristic methods such as EAs, GA and PSO are well-suited for handling such issues, they have been applied into the studies of MOPs and got many progresses in recent decades. Particularly, the PSO has attracted much attention in theory and applications since first proposed in [3]. Coello et al. in [4,5] applied an elite set to store the found optimal solutions, and used these solutions to guide other flying particles. Meanwhile, the search space is divided by grids to improve diversity. Parsopoulos and Michael present the weights polymerization method in [6]. Hu et al. proposed that they utilize a dynamic neighbor PSO algorithm to approach various optimization objectives in [7,8]. Ray and Liew combined the Pareto ranking mechanism and PSO algorithm together in [9], and it produced the non-dominated set through Pareto sorting and chose the best global particle by roulette. Pang et al. [10] solved the premature convergence problem by introducing a new density assessment scheme on particles' entropy information, and by adopting adaptive chaotic mutation operator, the MOPSO solutions emerge with good diversity and distribution. Chen et al. [11] proposed MOPSOEO algorithm based on PSO and external optimization (EO), which takes full advantage of the exploration ability of PSO and EO that overcome the problem of premature convergence for PSO when applied to MOPs.

In application areas, many single and multiple objective engineering problems have been solved by using PSO [1]. Roberge et al. [12] used PSO and genetic algorithm (GA) to cope with the complex computation of feasible and quasi-optimal trajectories for fixed wing UAVs in 3D environment. Xue et al. [13] present the first study on MOPSO for feature selection, and by introducing nondominated sorting, crowding, mutation, and dominance into PSO, the feature selection problems and Pareto front solutions are addressed. Zheng et al. [14] proposed an effective MOPSO method for population classification in fire evacuation operations, which simultaneously optimizes the precision and recall measures of the classification rules. Izzo et al. [15] adopted the constraint handling technique and multi-objective methods for PSO into the optimization problem of interplanetary trajectory.

However, for constrained multi-objective PSO (CMOPSO) algorithms, there is much less research especially compared with other algorithms such as EAs and GAs. This is partly due to the earlier establishment and more popularity for EAs and GAs than other optimization algorithms. On the other hand, it is partly because the optimization problems show to be more time complexity and algorithm complexity for constrained multi-objective situations, which cause unusual study difficulties. Ji introduced a symbiotic mechanism in [16] where the feasible particles evolve towards the front, and the infeasible particles evolve toward the feasible direction based on a feasible function. Reddy and Kumar proposed an EM-MOPSO algorithm that combined the PSO algorithm and Pareto dominance in [17]. Li et al. put forward an improved constrained multi-objective PSO algorithm based on the concept of constrained dominance, and disturb particles with small probability to enhance the diversity in papers [18,19]. Another constrained PSO algorithm, proposed by Worasucheep in [20], kept a stagnation detection mechanism that can automatically detect evolutionary a standstill state, and improve the dispersion of particles by the corresponding mechanism. Yen and Leong [21] proposed RCVMOPSO algorithm that utilized information of particles' infeasibility and feasibility status to search for feasible solutions, and constraints are converted into unconstrained objectives and handled by Pareto dominance relation.

The key procedures, for CMOPSO algorithms designing, are handling constraints and multi-objective functions. Fortunately, for these years the MOEAs have witnessed a large number of paper published on constraint handling techniques and multi-objective handling techniques which can be considered reasonably referenced for CMOPSO algorithms studies. Given this justification, the other population-based methods in MOEAs will be surveyed to review the developments of the MOPs.

MOEAs solving for MOPs have been evolved for decades, experiencing the traditional weight-sum aggregation approach, elitist Pareto-based approach and indicator-based algorithms, respectively [22,23]. In the early period, the weight-sum aggregation approach gets widely applications because of its simplicity. However, if objectives are conflicting with each other, it will cause solutions biasing towards one of the objectives [24,25]. For this consideration, in the late 1990s, Pareto based techniques are attracting much attention, and by using Pareto dominance relation and Pareto ranks for fitness assignment instead of fitness score, the improved solutions are achieved more than that of weighted sum approaches. The most representative elitist MOEAs include PAES [26], PESA [27] and PESA- [28,29], SPEA2 [30], NPGA2 [31], and NSGA-II [32], MOEA/D [33]. More recently, the indicator-based algorithm [34], such as the S metric selection evolutionary multi-objective optimization algorithm (SMS-EMOA) [35], caught a new trend which performs better in the presence of many objectives.

On the other hand, various constraints handling techniques targeted at EAs have been developed to solve CMOPs [36]. Coello and Christiansen [37] proposed two new MOEAs based on the concept of min-max optimum, but they only optimize feasible solutions since only feasible solutions can survive to the next generation. Deb [31] introduced a constrained domination principle to handle constraint in NSGA-II. By this principle, all individuals can be ranked through Pareto dominance relationship and constraint violations. Due to this advantage, this technique later is widely used in microgenetic algorithm (micro GA) and MOPSOs [38,39]. Ray and Won [40] also employ standard min-max formulation for constraint handling and divide the objective space into a predefined number of radial slots where the solutions will compete with members in the same slot for existence. Geng et al. [41] introduced the strategy of infeasible elitists to act as a bridge connecting any isolated feasible regions during the evolution process, which appears significant improvement in distributions and quality of the Pareto fronts. Harada et al. [42] proposed Pareto descent repair (PDR) operator to repair the infeasible solution that aims to reduce all violated constraints simultaneously. To overcome the parameter tuning problem for single constraint handling technique, Qu and Suganthan [43] proposed an ensemble of constraint handling methods (ECHM) to tackle constrained multi-objective optimization problems.

Motivated by this research background, in this paper, we proposed a hybrid constrained MOPSO algorithm based on adaptive penalty approach and normalized non-dominated sorting approach to solve the CMOPs. The organization of this paper is as follows. In the second section of this paper, we describe the constrained multi-objective optimization problem in general form, basic principle of the PSO algorithm and necessary concepts. In Section 3, algorithm design key issues are elaborated including constraint handling technique, multi-objective handling technique, and external population update mechanism. In Section 4, the simulation results of the proposed algorithm are provided with respect to four typical test problems. In Section 5, an application example of hypersonic reentry trajectory optimization problem has been solved by the proposed algorithm. Finally, Section 6 is a summary of the full article.

2. View of Constrained MOP and PSO Algorithm.

2.1. General constrained multi-objective optimization problem (CMOP).
Without loss of generality, only the minimization problems will be assumed, and the constrained multi-objective optimization problem can be described as follows:

$$\begin{array}{rl} Minimize: & f(x) = (f_1(x), f_2(x), \ldots, f_l(x)) \\ s.t. & g_j(x) \le 0,\ j = 1, 2, \ldots, q \\ & h_j(x) = 0,\ j = q+1, q+2, \ldots, m \\ & x = (x_1, x_2, \ldots, x_n), \\ & x_i^{\min} \le x_i \le x_i^{\max},\ i = 1, 2, \ldots, n \end{array} \tag{1}$$

where l is the number of objective functions, x is decision variable, $x_i^{\min}$ and $x_i^{\max}$ are upper and lower bounds of each dimension of the decision variables, $i = 1, 2, \ldots, n$, $g_j(x)$ and $h_j(x)$ are both n-ary functions on R^n, $f(x) = (f_1(x), f_2(x), \ldots, f_l(x))$ is the objective function, $g_j(x)$ is the j-th inequality constraints, and $h_j(x)$ is the j-th equality constraints.

2.2. Particle swarm optimization.
Particle swarm optimization (PSO) [3], proposed by Kennedy and Eberhart in 1995, has been successfully applied in many optimization problems because of its simple principle and easy implementation. So far, it has achieved profound development, and gradually becomes more significant in research for solving CMOPs.

In standard PSO, the velocity and position of particle i in the search space are calculated based on the following equation:

$$\begin{cases} v^i(t+1) = \omega v^i(t) + c_1 r_1()[Pbest^i - x^i(t)] + c_2 r_2()[Gbest - x^i(t)] \\ x^i(t+1) = x^i(t) + v^i(t+1) \end{cases} \tag{2}$$

where, $v^i(t+1)$ is the velocity of particle i in generation $t+1$, $x^i(t+1)$ is the position of particle i in generation $t+1$, $Pbest^i$ is the current optimal position of particle i, and $Gbest$ is current global optimal position. ω is the inertia coefficient, $r_1()$ and $r_2()$ are two random numbers with uniform distribution on the interval $[0, 1]$, and c_1 and c_2 are acceleration factors, which represent the weights of each particle being pushed towards the statistical Pbest and Gbest position, respectively.

2.3. Related definitions.

2.3.1. *Pareto dominance.* A solution $u = (u_1, u_2, \ldots, u_n)$ is said to Pareto-dominate solution $v = (v_1, v_2, \ldots, v_n)$, if and only if $f_i(u) \le f_i(v)$ $(i = 1, 2, \ldots, l)$, and there exists at least one $j \in \{1, 2, \ldots, l\}$ that satisfies $f_j(u) < f_j(v)$, which are denoted by $u \prec v$, and referred to as u dominate v.

2.3.2. *Pareto-optimal solution.* A solution is said to be a Pareto-optimal solution if and only if there exists no v and the feasible region allows that $v \prec u$.

2.3.3. *Constrained-domination.* A solution i is said to constrained-dominate a solution j, if any of the following conditions is true [32]: 1) Solutions i and j are both feasible solutions, and solution i dominates solution j. 2) Solution i is feasible and solution j is not. 3) Solutions i and j are both infeasible, but solution i has a smaller constraint violation. 4) Solutions i and j are both infeasible, constraint violation of solution i equals that of solution j, and solution i dominates solution j.

2.3.4. *Crowding distance.* The crowding distance, first proposed in [32], is the sum of the average distance of two points on either side of this point along each of the objective's dimension.

Figure 1 shows that the crowding distance of point i can be calculated by

$$\frac{|f_1(x_{i+1}) - f_1(x_{i-1})|}{f_{1\max} - f_{1\min}} + \frac{|f_2(x_{i+1}) - f_2(x_{i-1})|}{f_{2\max} - f_{2\min}} \tag{3}$$

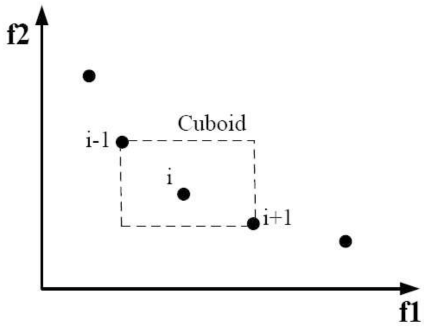

FIGURE 1. The crowding distance calculation

2.3.5. *Generational distance (GD).* Generational distance (see [44]) is the distance between non-dominated solutions and the Pareto-optimal solutions. The calculation equation is as follows:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{4}$$

where, n is the number of non-dominated solutions, d_i is the minimum distance between the i-th solution to the Pareto optimal solution set. Especially, a value of 0 indicates that all the individuals generated are in the Pareto optimal set. This metric reflects the approaching level of the non-dominated solutions to the Pareto optimum set.

2.3.6. *Spacing (SP).* Spacing [45] is the metric desiring to measure the spread (distribution) of vectors throughout the non-dominated vectors found so far, and the metric reflects the diversity of the resulting front. The calculation equation is as follows:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(\bar{d} - d_i\right)^2} \tag{5}$$

where, $d_i = \min\limits_{\substack{j=1,2,\ldots,n \\ j \neq i}} \left(\sum_{k=1}^{l} |f_k(x_i) - f_k(x_j)| \right)$, $i = 1, 2, \ldots, n$, denotes the distance between the objective vector of the non-dominated solution x_i and its nearest objective vector. $\bar{d}$ is the average of d_i. SP $= 0$ means the corresponding front of the non-dominated solutions complete diverse distribution.

3. **Description of the Proposed Approach.** The main principle for solving CMOPs is to convert the constraints into unconstrained multi-objective problems, so an effective constraint handling mechanism design is considered as the key. Recently, the penalty function method is widely applied for constraint handling, in order to avoid dealing with too many penalty parameters. In this section, a typical adaptive constraint handling method is described, combined to a fast normalized non-dominated sorting technique and

the improved external non-dominated population maintenance according to the elitist strategy and constrained dominance.

3.1. **A constraint handling technique: adaptive penalty method.** Woldesenbet, Tessema and Yen in [46] provide a method for handling constraints in MOPs called adaptive penalty, which possesses good versatility. This approach makes the combination of objective function value and individual constraint violation, and defines distance measurement and adaptive penalty functions, whose values are calculated by individual feasibility and constraint violations, and thus by using this method the new objective function is constructed.

The new objective function $F_i(x)$ is represented according to the following equation

$$F_i(x) = d_i(x) + p_i(x) \quad i = 1, 2, \dots, l \tag{6}$$

Equation (6) is comprised of the distance function $d_i(x)$ and the penalty function $p_i(x)$, $i = 1, 2, \dots, l$, where the distance function $d_i(x)$ is defined as follows,

$$d_i(x) = \begin{cases} v(x) & \text{if } \gamma_f = 0 \\ \sqrt{\tilde{f}_i(x)^2 + v(x)^2} & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, l \tag{7}$$

In Equation (7), normalized objective functions $\tilde{f}_i(x)$ are defined as follows:

$$\tilde{f}_i(x) = \frac{f_i(x) - f^i_{\min}}{f^i_{\max} - f^i_{\min}} \quad i = 1, 2, \dots, l \tag{8}$$

where $f^i_{\max} = \max\limits_x f_i(x)$ and $f^i_{\min} = \min\limits_x f_i(x)$ indicate the maximum value and the minimum value of the objective function in the i-th dimension of the objective space, respectively. In Equation (7), individual constraint violation $v(x)$ is defined as follows:

$$v(x) = \frac{1}{m} \sum_{j=1}^{m} \frac{c_j(x)}{c_j^{\max}} \tag{9}$$

where,

$$c_j(x) = \begin{cases} \max(0, g_j(x)) & \text{if } j = 1, 2, \dots, q \\ \max(0, |h_j(x)| - \delta) & \text{if } j = q+1, q+2, \dots, m \end{cases} \tag{10}$$

δ is a small positive number (usually 0.001 or 0.0001), that denotes the tolerance value for equality constraints, and constraint violation $v(x)$ is zero when x is a feasible solution. In Equation (7), the proportion of feasible solutions in population γ_f is

$$\gamma_f = \frac{\text{the number of feasible individuals in current population}}{\text{population size}} \tag{11}$$

In Equation (6), penalty function $p_i(x)$ is defined as follows:

$$p_i(x) = (1 - \gamma_f) X(x) + \gamma_f Y_i(x) \tag{12}$$

where, $X(x) = \begin{cases} 0 & \text{if } \gamma_f = 0 \\ v(x) & \text{otherwise} \end{cases}$, $Y_i(x) = \begin{cases} 0 & \text{if } x \text{ is feasible} \\ \tilde{f}_i(x) & \text{otherwise} \end{cases}$.

It can be observed that new objective function value of each individual can be calculated by Equations (6)-(12). This approach automatically adjusts individuals' penalties with the proportion of feasible solutions in the population and the individual constraint violation, which ensures only infeasible solutions can be punished. The greater infeasible individual violations constrain, the greater penalty it will be affected. Furthermore, this technique avoids the introduction of penalty parameters, and the new objective function provides outstanding adaptability for all feasible and infeasible solutions. Therefore, the algorithm analytical and calculation complexity are effectively reduced. Additionally, the objective functions with form (6) can be directly used for multi-objective particle swarm optimization operations which will be described in Section 3.2.

3.2. **Normalized non-dominated sorting.** Bao and Zhu proposed a normalized sorting method [47] that ranks the non-dominated individuals as a sequence in a set. Its run-time complexity is demonstrated to be not more than $O(n \log n) + O(lnm)$ (n is the population size, m is the number of non-dominant solutions, and l is the number of objects), which is less than that of the classical approach NSGA-II $O(ln^2)$. This advantage provides a new application prospect for this approach. The basic steps are as follows.

Normalizing the objective function $F_i(x)$ is obtained from constraints handling in Section 3.1, by the following equation:

$$\overline{F}_i(x) = \frac{F_{i\max}(x) - F_i(x)}{F_{i\max}(x) - F_{i\min}(x)} \tag{13}$$

Sum all the normalized functions on various dimensions of objective space, so the normalized mixed function is obtained:

$$G(x_j) = \sum_{i=1}^{r} \overline{F}_i(x_j), \quad i = 1, 2, \ldots, l, \ j = 1, 2, \ldots, N \tag{14}$$

where, l is the number of objective functions, and N is population size.

Sort the normalized mixed function in descending order,

$$G(u_1) \geq G(u_2) \geq \cdots G(u_N) \tag{15}$$

All individuals are stored into an array $S[N]$, that is $S[1] = u_1$, $S[2] = u_2, \ldots, S[N] = u_N$, and $u_1 \prec= u_2 \prec= \cdots \prec= u_N$ is the result of the non-dominated sorting of all the individuals in the population, where $\prec=$ denotes a dominate position.

In sense of this sorting, it can be proved that the individual in front will (at least) not be dominated by the following individual, while the following individual will also possibly not be dominated by the previous one. Therefore, under normal circumstances, we can consider that the individuals in front sequence with better ranks are superiorly dominant individuals.

3.3. **External archive maintenance.** Individuals and their new objective function (6), after being handled by adaptive penalty technique, rank according to the normalized non-dominated sorting technique, which implies individuals with better dominance have higher priorities over those with poor dominance. Each new individual generated by PSO algorithm produces the N individuals sequence after operations of the constraints handling and summation of the normalized functions. Then the chosen non-dominated solutions are used for updating the external archive (an external non-dominated elite population).

All the sorted individuals have been compared with individuals in external elite population in paper [47], and the one not dominated by the external individuals will be chosen into the external elite population for updating. In fact, since the size of the external population is often far less than that of internal population, it seems unnecessary to compare all the internal population individuals with elite individuals for external population updating. Furthermore, individuals with better rank can be considered as individuals with the better dominance in the internal population, because the worse ranking individuals certainly could not dominate their previous individuals. Therefore, in this paper, we use the first M better order individuals (M is the external population size) to update the external population, and the specified update mechanism is as follows.

If the external population is not full, copy the individuals directly into external population; if the external population is full, compare the new individuals i with each of the external individuals j, 1) if individual j dominates individual i, do not copy in; 2) if individual i dominates only one external individual j, replace individual j with individual i; if individual i dominates more than one external individual, eliminate all these external individuals, and copy individual i in; 3) if individual i does not dominate any external individual, and is not dominated by any external individual, then add individual i into the external population, and calculate the crowding distance (see Section 2.3.4) of M + 1 individuals. Then remove the individual with the smallest crowding distance from the external population.

It must be noted that, individuals are compared with each other by the relationship of Pareto constrained-dominance. Furthermore, although sometimes the better ranking individuals cannot dominate the worse ranking individuals, this operation would still greatly improve the speed and efficiency of the proposed algorithm, as it is unnecessary to compare every individual with external individuals. In Section 4, simulations will prove this approach optimization results are good.

3.4. **Global best update.** For PSO algorithm, it is clear that the global optimal particle should be selected from the external archive. In this paper, the following operation is taken to obtain it.

For a certain generation, calculate the crowding distance of each individual in external population. 1) If the crowding distances of individuals are all infinite, select one individual randomly as Gbest; 2) if there is a finite number, choose the individual that possesses the largest crowding distance as Gbest.

3.5. **Program flowchart of the proposed algorithm.** From the aforementioned analysis, the process of the PSO algorithm for solving CMOPs problems is executed as Figure 2.

4. **Simulation Results.** In this section, the newly proposed algorithm is tested on four different test problems with performance metrics for convergence (Generational distance, GD), distribution (Spacing, SP) and algorithm running time (Elapsed time).

Four typical test problems (see Table 1) are chosen for performance tests, and then performance results are compared between the newly proposed CMOPSO and the classical algorithm in [46].

Table 1. Constrained test problems used in this study

Problem	n	Variable bounds	Objective functions	Constraints
CONSTER	2	$x_1 \in [0.1, 1]$ $x_2 \in [0, 5]$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = (1 + x_2)/x_1$	$g_1(\vec{x}) = -9x_1 - x_2 + 6 \leq 0$ $g_2(\vec{x}) = -9x_1 + x_2 + 1 \leq 0$
SRN	2	$x_i \in [-20, 20]$ $i = 1, 2$	$f_1(\vec{x}) = (x_1 - 2)^2 + (x_2 - 1)^2 + 2$ $f_2(\vec{x}) = 9x_1 - (x_2 - 1)^2$	$g_1(\vec{x}) = x_1^2 + x_2^2 - 225 \leq 0$ $g_2(\vec{x}) = x_1 - 3x_2 + 10 \leq 0$
TNK	2	$x_i \in [0, \pi]$ $i = 1, 2$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = x_2$	$g_1(\vec{x}) = -x_1^2 - x_2^2 + 1 + 0.1\cos(16\arctan(x_1/x_2)) \leq 0$ $g_2(\vec{x}) = (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0$
BINH4	2	$x_i \in [-10, 10]$ $i = 1, 2$	$f_1(\vec{x}) = 1.5 - x_1(1 - x_2)$ $f_2(\vec{x}) = 2.25 - x_1(1 - x_2^2)$ $f_3(\vec{x}) = 2.625 - x_1(1 - x_2^3)$	$g_1(\vec{x}) = -x_1^2 - (x_2 - 0.5)^2 + 9 \leq 0$ $g_2(\vec{x}) = (x_1 - 1)^2 + (x_2 - 0.5)^2 - 6.25 \leq 0$

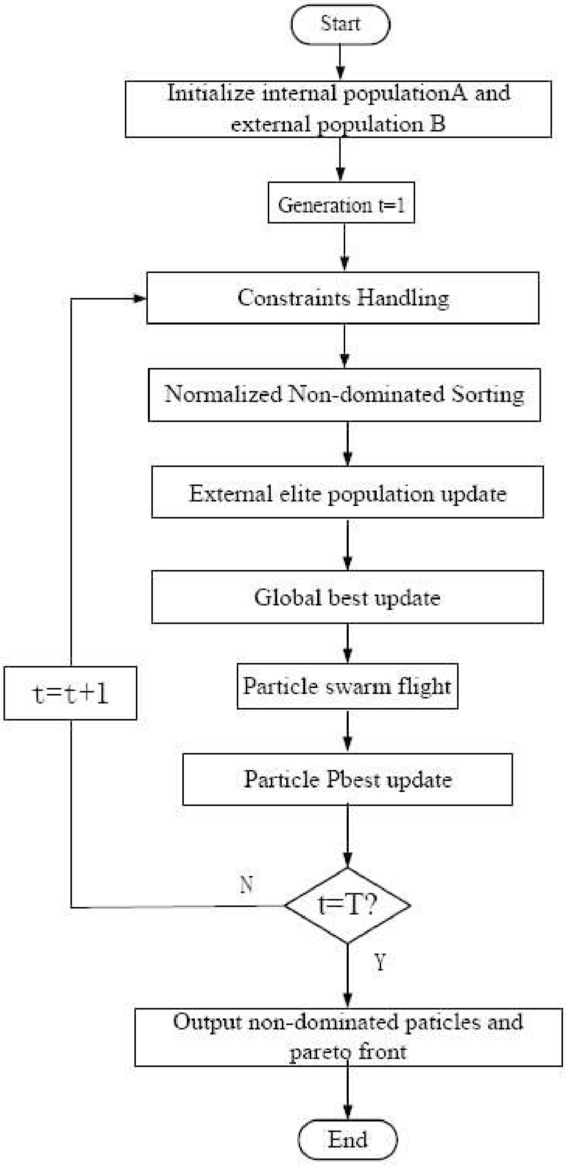

Figure 2. Program flowchart of the proposed algorithm

Table 2. GD metric of the proposed algorithm on CONSTER, SRN and TNK

	CONSTER	SRN	TNK
Algorithm compared	0.0191	0.0168	0.0099
Proposed algorithm	0.0025	0.0012	0.0023

In this test, we set population size for 100 (200 for problem BINH4), external population size of 30 (50 for BINH4), and a maximum generation number of 300.

The CONSTER test problem was proposed in [53]. As shown in Figure 3, it can be seen that 30 non-dominated points are evenly distributed on the true Pareto front. From Table 2, the convergence metric GD of the Pareto front is 0.0025 which is significantly smaller than that of the algorithm compared. Moreover, the metric of SP (equal to 0.1165) is also better than the algorithm compared. For this function, the average time of the proposed algorithm independently running 30 times is 5.7851 seconds. The SRN test problem is tested with the population of 100, generation of 100, and 30 non-dominant

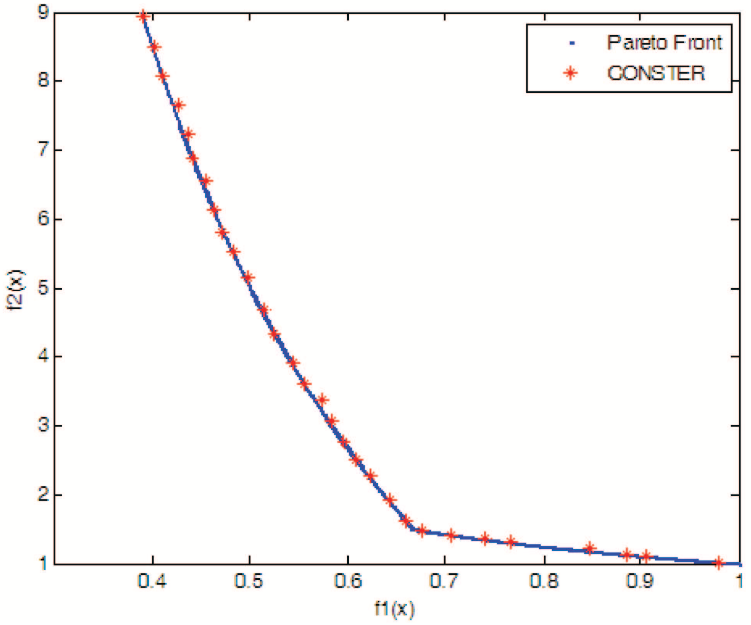

FIGURE 3. Pareto front obtained with the proposed algorithm on problem CONSTER

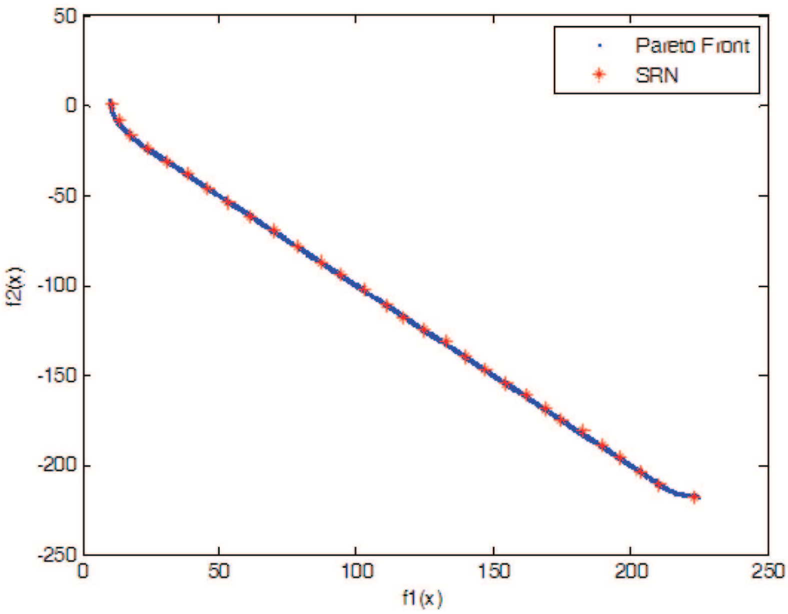

FIGURE 4. Pareto front obtained with the proposed algorithm on problem SRN

individuals. As can be compared, diversified non-dominant individuals of the proposed algorithm distributed on the Pareto front, whose excellent convergence and distribution performance can also been reflected by GD metric (0.0012) and SP metric (0.074) hold on the true Pareto front. The average time of the proposed algorithm independently running 30 times is 7.1198s. The TNK function [54] results are shown in Figure 5, and the proposed algorithm provides feasible optimal solutions that are diversely distributed on true Pareto front. The non-dominated solutions converge uniformly towards the discontinuous Pareto front and cover the whole extent of the Pareto front. From Tables 2 and 3, GD (0.0023) and SP (0.0125) of the proposed algorithm show superiority to the algorithm compared. This test used the population size 100, with 30 external non-dominated solutions, and 300

TABLE 3. SP metric of the proposed algorithm on CONSTER, SRN and TNK

	CONSTER	SRN	TNK
Algorithm compared	0.3210	0.385	0.4660
Proposed algorithm	0.1165	0.074	0.0125

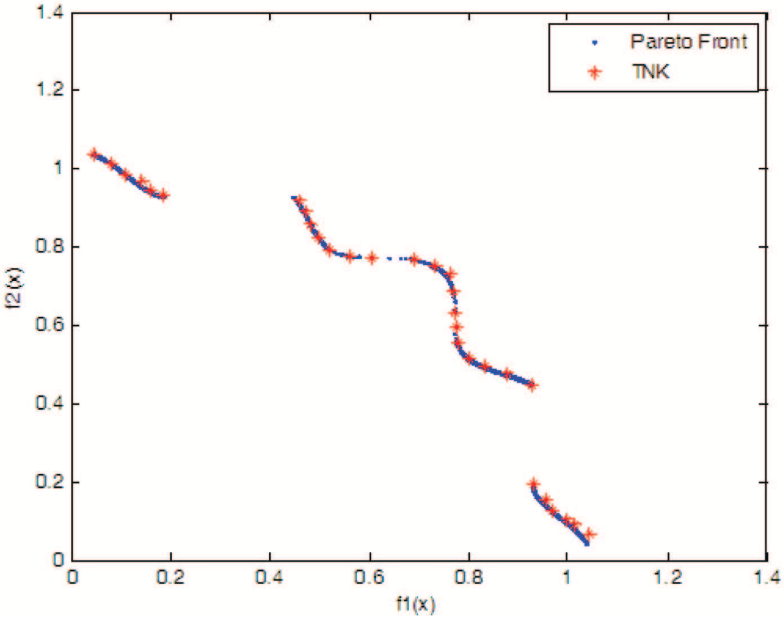

FIGURE 5. Pareto front obtained with the proposed algorithm on problem TNK

generations. Generations are improved because the noncontiguous front requires bigger iterations to reach the global optimum. The average time of the proposed algorithm independently running 30 times is 7.3302 seconds.

The BINH4 test adopted population size 300, 50 external non-dominated solutions, and 300 generations. The final Pareto front optimized by the proposed algorithm is drawn in Figure 6. On the single value segment, non-dominated solutions accurately converge to the Pareto front, while on the multi-value segment, non-dominated solutions converge to the Pareto front as a whole. Running independently 30 times the average of GD, SP indicators and the running time are 0.0151, 0.1612, and 30.9691 seconds, respectively.

From the above simulation, the proposed CMOPSO algorithm shows precise convergence and diverse distribution on the Pareto front. Compared with the data in article [46], the algorithm we propose possesses better characteristics of GD and SP than the former, which shows the advantages of this algorithm.

5. **Application on Hypersonic Reentry Glide Vehicles (HRGVs) Trajectory Optimization.**

5.1. **Description for HRGVs trajectory optimization problem.** To further verify the CMOPSO algorithm proposed in previous sections, a trajectory optimization problem for Hypersonic Reentry Glide Vehicles (HRGVs) will be applied in this section. The hypersonic reentry glide vehicle belongs to a complex system with characteristics of highly nonlinear, strong coupling and fast time-varying, whose reentry trajectory optimization project can be constructed as an optimization problem with multiple constraints and multiple objects. The trajectory optimization problem aims to seek optimal flight trajectory (or trajectories) that guarantee specified performance as well as satisfying the constraints

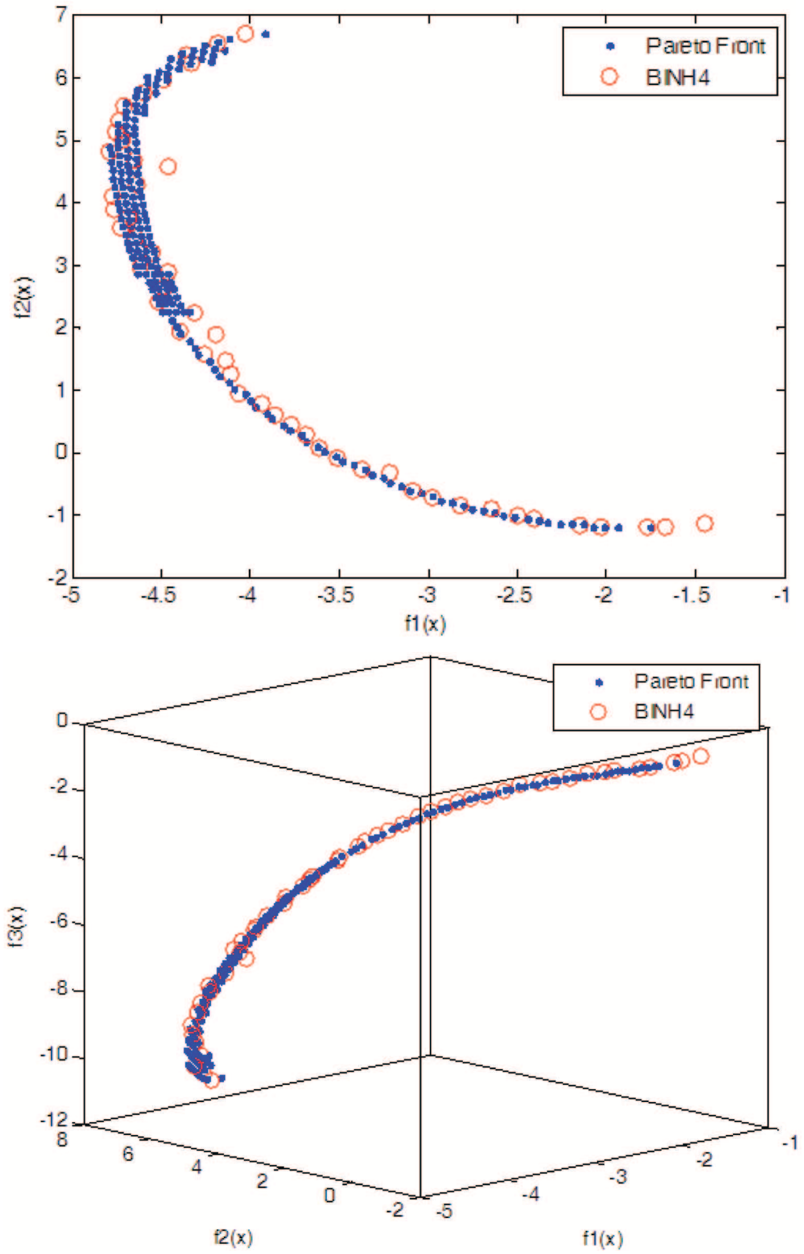

Figure 6. Pareto front obtained with the proposed algorithm on problem BINH4 (top: top view, down: 3D view)

such as heat peak, dynamic pressure and aerodynamic load factor for appointed vehicle flight missions.

Chen et al. [55] introduced the NSGA-II algorithm into the design of RLV multi-objective reentry optimization with minimum heat and maximum maneuverable range. Xie et al. [56] presented a Migrant PSO algorithm to solve the trajectory optimization which proves to be able to generate an optimal 3DOF reentry trajectory rapidly. Jiao and Jiang [57] proposed the colony algorithm method for multi-objective optimization of reentry trajectory planning for hypersonic aircraft. Zhao and Zhou [58] studied the end-to-end trajectory optimization problem by single objective function and multiple constraints based on constrained PSO for hypersonic reentry vehicles.

As the reentry trajectory optimization problem for HRGVs presents to be an essentially constrained multi-objects issue, traditional gradient-based methods have much difficulty in solving the multi-objectives. Therefore, the application of CMOPSO shows high practical significance. According to these motivations, in this section, the CMOPSO framework will be considered into the problem solving. The 3DOF point-mass dynamics of the vehicle are described by the following dimensionless equations of motion:

$$\begin{cases} \frac{d\tilde{R}}{de} = \tilde{V}\sin\gamma\left(\tilde{V}\tilde{D} - \tilde{V}\varphi_{\tilde{V}3}\right)^{-1} \\ \frac{d\varphi}{de} = \frac{\tilde{V}\cos\gamma\cos\chi_0}{\tilde{R}}\left(\tilde{V}\tilde{D} - \tilde{V}\varphi_{\tilde{V}3}\right)^{-1} \\ \frac{d\lambda}{de} = \frac{\tilde{V}\cos\gamma\sin\chi}{\tilde{R}\cos\varphi}\left(\tilde{V}\tilde{D} - \tilde{V}\varphi_{\tilde{V}3}\right)^{-1} \\ \frac{d\tilde{V}}{de} = \left[-\tilde{D} - \frac{\sin\gamma}{\tilde{R}^2} + \varphi_{\tilde{V}3}\right]\left(\tilde{V}\tilde{D} - \tilde{V}\varphi_{\tilde{V}3}\right)^{-1} \\ \frac{d\gamma}{de} = \frac{1}{\tilde{V}}\left[\tilde{L}\cos\mu + \left(\tilde{V}^2 - \frac{1}{\tilde{R}}\right)\frac{\cos\gamma}{\tilde{R}} + \varphi_{\gamma_3} + \varphi_{\gamma_4}\right]\left(\tilde{V}\tilde{D} - \tilde{V}\varphi_{\tilde{V}3}\right)^{-1} \end{cases} \tag{16}$$

$$\begin{aligned} \varphi_{\tilde{V}_3} &= \tilde{\omega}_e^2\tilde{R}\cos\varphi\left(\cos\varphi\sin\gamma - \sin\varphi\cos\chi_0\cos\gamma\right) \\ \varphi_{\gamma_3} &= 2\tilde{\omega}_e\tilde{V}\cos\varphi\sin\chi_0 \\ \varphi_{\gamma_4} &= \tilde{\omega}_e^2\tilde{R}\cos\varphi\left(\cos\varphi\cos\gamma + \sin\varphi\cos\chi_0\sin\gamma\right) \\ \varphi_{\chi_3} &= 2\tilde{\omega}_e\tilde{V}\left(\sin\varphi - \cos\varphi\cos\chi_0\tan\gamma\right) \\ \varphi_{\chi_4} &= \frac{\tilde{\omega}_e^2\tilde{R}}{\cos\gamma}\cos\varphi\sin\varphi\sin\chi_0 \end{aligned}$$

where, $\tilde{R} = \frac{R}{R_0}$, $\tilde{V} = \frac{V}{\sqrt{g_0R_0}}$, $\tilde{\omega}_e = \frac{\omega_e}{\sqrt{g_0/R_0}}$, $\tilde{D} = \frac{D}{Mg_0}$, $\tilde{L} = \frac{L}{Mg_0}$, $\tau = \frac{t}{\sqrt{R_0/g_0}}$, and R, V, ω_e, D, L, t are the radial distance from the center of the Earth to CAV, Earth-relative velocity, Earth self-rotation rate, drag force, lift force and entry time respectively. And $\tilde{R}$, $\tilde{V}$, $\tilde{\omega}_e$, $\tilde{D}$, $\tilde{L}$, τ are the corresponding dimensionless forms of R, V, ω_e, D, L and t. φ denotes the latitude and λ the longitude. The flight path angle is γ and μ the bank angle. The velocity azimuth angle χ is measured from the North in a clockwise direction. Energy-like variable e is defined as $e = \frac{1}{\tilde{R}} - \frac{1}{2}\tilde{V}^2$.

Typical reentry trajectory inequality path constraints include path constraints, Terminal Constraints and Control Constraints shown as follows

$$\dot{Q} = \frac{C_1}{\sqrt{R_d}}\sqrt{\rho}V^{3.15} \leq \dot{Q}_{\max} \tag{17}$$

$$n = \frac{\sqrt{L^2 + D^2}}{Mg_0} \leq n_{\max} \tag{18}$$

$$q = \frac{1}{2}\rho V^2 \leq q_{\max} \tag{19}$$

$$R(t_f) \in [R_f - \Delta R_{down}, R_f + \Delta R_{up}] \tag{20}$$

$$V(t_f) \in [V_f - \Delta V_{down}, V_f + \Delta V_{up}] \tag{21}$$

$$\gamma(t_f) \in [\gamma_f - \Delta\gamma_{down}, \gamma_f + \Delta\gamma_{up}] \tag{22}$$

$$\alpha \in [\alpha_{\min}, \alpha_{\max}] \tag{23}$$

where Equation (17) is a constraint on the heating rate at a specified point on the surface of the hypersonic vehicle, with the constant $C_1 = 11093$ and curvature radius of the stagnation point $R_d = 0.01$m. Equation (18) is a constraint on the total aerodynamic load factor on the body of the hypersonic vehicle ($n_{\max}$ is in the unit of g_0). The constraint Equation (19) is on the dynamic pressure ($q_{\max}$ is in the unit of N/m^2).

The control variables of the longitudinal entry motion are angle of attack α, which lies in lift L and drag D through the lift and drag coefficient C_L and C_D. In this paper, we set the longitudinal average margin value of the bank angle to $\pi/6$.

Since the main trajectory control variable is angle of attack α, the optimization problem purpose is to achieve the optimal angle of attack. In reentry engineering, the angle of attack is generally adopted to be a parametric form as shown in Formula (24), so, decision variables of this problem are transformed to be angle of attack parameters V_1, V_2.

$$\alpha = \begin{cases} \alpha_{\max} & V \in (V_1, V_0] \\ \frac{\alpha_{\max L/D} - \alpha_{\max}}{V_2 - V_1}(V - V_1) + \alpha_{\max} & V \in (V_2, V_1] \\ \alpha_{\max L/D} & V \in (V_f, V_2] \end{cases} \tag{24}$$

The objective functions of the reentry trajectory optimization problem vary with the missions and reference indicators, and usually not single and changeless. Some typical objective functions are shown as follows:

$$f_1 = S(e_f) = \mathrm{Re} \cdot \cos^{-1}[\cos\varphi_0 \cos\varphi \cos(\lambda_0 - \lambda) + \sin\varphi_0 \sin\varphi] \tag{25}$$

$$f_2 = \int_{e_0}^{e_f} |\dot{\gamma}|\, de \tag{26}$$

where, $S(e_f)$ in Formula (25) denotes maximizing gliding range, defined as the circle distance on the surface of spherical Earth from the vehicle position (λ_0, φ_0) to the terminal position (λ, φ). Formula (26) denotes minimizing total ballistic oscillation.

5.2. **Principle for HRGVs trajectory optimization.** Based on the proposed CMOPSO algorithm, the trajectory optimization problem can be solved by the following steps.

As a detailed discussion for the proposed CMOPSO algorithm has been made in previous sections, in this application, this method will be directly embedded into the HRGVs trajectory optimization principle shown in Figure 7, and complying with the principle, the main steps are stated as follows:

Step 1: Initialize the first population;

Step 2: Calculate the reentry trajectories of the particle swarm with Equation (16). If the constraints satisfy error tolerances, shift to Step 5. Otherwise, shift to Step 3;

Step 3: Optimize the particle swarm with the proposed CMOPSO algorithm, including constraints handling, non-dominated sorting, and external archive population update and particle swarm flight;

Step 4: Judge if the algorithm achieves the max iteration times T, if not, return to Step 2. Otherwise, shift to Step 5;

Step 5: Output the external archive and multi-objective values.

5.3. **Optimization simulation.** In this simulation, the path constraints parameters are $\dot{Q}_{\max} = 2000\mathrm{kW/m}^2$, $n_{\max} = 2$, $q_{\max} = 50\mathrm{kN/m}^2$. Three sets of angle of parameters are set in this problem, which means three trajectories will be achieved after optimization. The algorithm iteration steps are set to 20, population size is 12, and non-dominant population size is 3. The initial value of the numerical integration variable for the 3DOF equations of motion is $e_0 = 0.567$, and its terminal value is $e_f = 0.969$.

Table 4 shows the initial values $X(e_0)$, desired terminal states values $X(e_f)$ for all the state variables during reentry period. After optimization, the error values of the terminal states should not exceed their tolerance ranges ΔX provided in Table 4.

Through the simulation, three groups of turning velocity have been optimized, they are respectively, $(6241.33, 3352.38)$, $(5046.63, 3204.32)$, $(5301.91, 3317.75)$ which construct three different reentry angle of attack profiles, as shown in Figure 8. Based on these angles of attack, three corresponding longitudinal reentry trajectories can be obtained, shown in

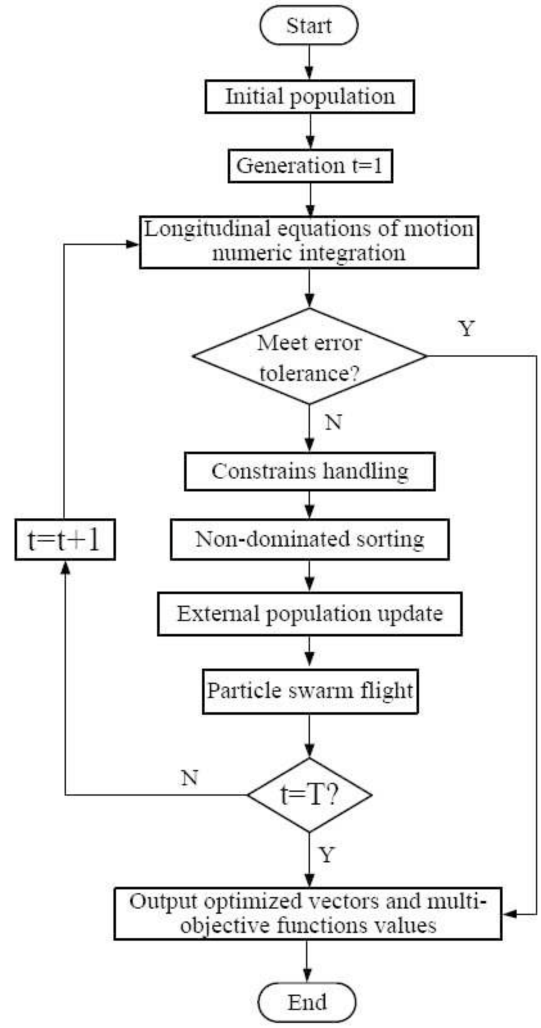

Figure 7. Flowchart of trajectory optimization problem using PSO algorithm

Table 4. Reentry conditions

	$X(e_0)$	$X(e_f)$	ΔX
H (km)	120	30	2
φ (deg)	0	–	–
λ (deg)	0	–	–
V (m/s)	7200	1800	10
γ (deg)	0	–7.5	1
χ (deg)	110	–	–

Table 5. Results of the trajectory optimization

	Trajectory 1	Trajectory 2	Trajectory 3
$H(e_f)$ (km)	28.783	29.887	29.418
$V(e_f)$ (m/s)	1806.56	1800.61	1803.14
$\gamma(e_f)$ (deg)	–7.37	–6.91	–7.15
$t(e_f)$ (sec)	1557.53	1720.91	1681.34
$S(e_f) = 1/f_2$ (km)	9752.117	10425.856	10282.587
f_4	11263.08	11273.54	11269.63

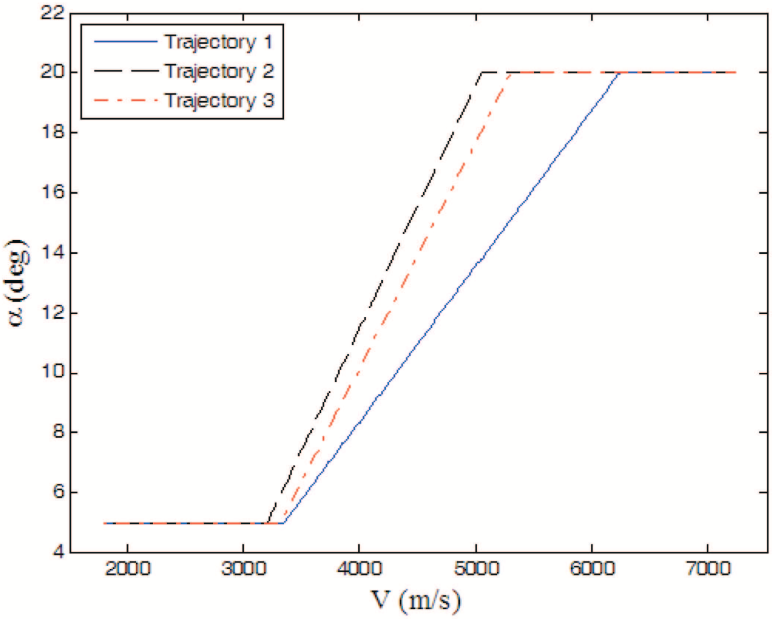

FIGURE 8. Angle of attack profile under three optimized parameters

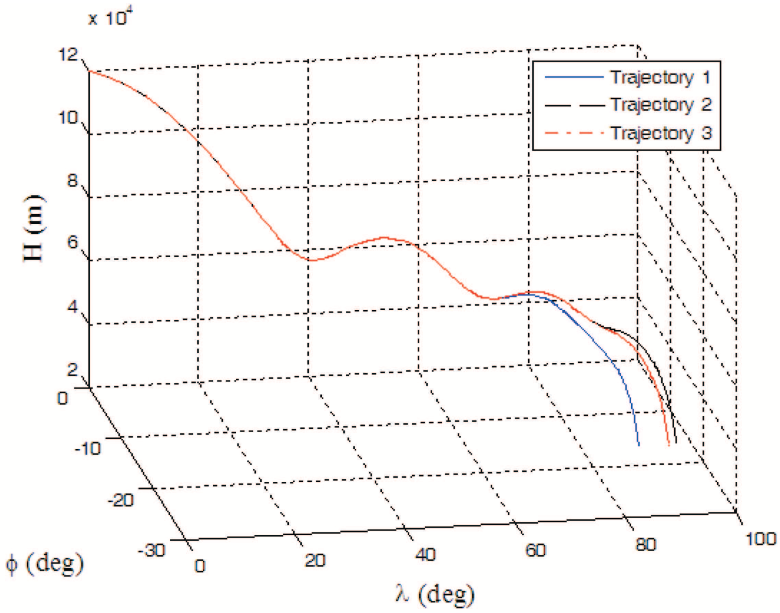

FIGURE 9. Three reentry trajectories under three angle of attack profiles

Figure 9. All the trajectory initial conditions and terminal conditions are listed in Table 4 and Table 5. And from Table 5 it can be seen that all the constrained terminal states values strictly met the error tolerance ranges, the objective functions f_1 and f_2 form a set of contradictions as any trajectory cannot be superior to any other trajectories based on the evaluation of these two objective functions. Figure 10 illustrates the situation of the three optimized trajectories satisfies the reentry constraints.

6. **Conclusions.** This article discusses the CMOPs solving with the frame of PSO. A CMOPSO algorithm is proposed which combines two approaches in CMOPs together, the constraint handling technique with adaptive penalty function, and the multi-objective handling technique with normalized non-dominated sorting. Furthermore, the update

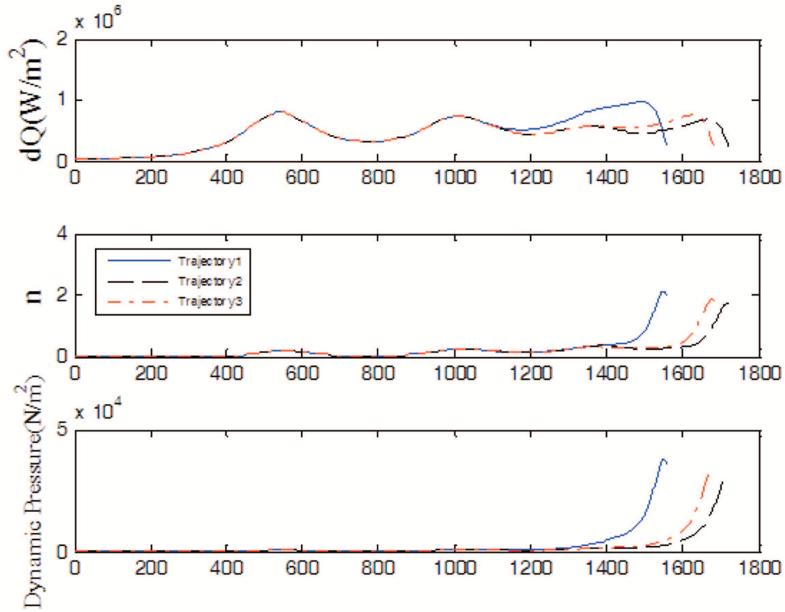

Figure 10. Reentry constraints histories of the three trajectories

strategy of the external elite archive is improved based on a novel sorting operation. All the above techniques are integrated with the basic PSO algorithm and construct a new algorithm to solve CMOPs.

The constraints handling technique takes adaptive penalty function and distance measurements into consideration which avoids introducing the penalty factor parameters while enhancing the adaptability of punishment. The normalized non-dominated sorting technique ranks all the normalized objective functions to a sequence with a superiorly dominated order. Experimental results including two pivotal performances show the efficiency of this improvement.

Test functions and trajectory optimization of a hypersonic reentry glide vehicle are simulated to demonstrate the effectiveness and good performance of the proposed algorithm and verify the application significance of the proposed CMOPSO algorithm.

Acknowledgment. This work is supported by the National Natural Science Foundation of China [61273095].

REFERENCES

[1] S. Lalwani, S. Singhal, R. Kumar et al., A comprehensive survey: Applications of multi-objective particle swarm optimization (MOPSO) algorithm, *Transactions on Combinatorics*, vol.2, no.1, pp.39-101, 2013.
[2] M. C. Fonseca and J. P. Fleming, Genetic algorithms for multi-objective optimization: Formulation discussion and generalization, *ICGA*, vol.93, pp.416-423, 1993.
[3] K. James, Particle swarm optimization, *Encyclopedia of Machine Learning*, pp.760-766, 2010.
[4] C. A. C. Coello and M. S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, *Proc. of the 2002 Congress on Evolutionary Computation*, vol.2, 2002.
[5] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Transactions on EVOL COMPUT*, vol.8, no.3, pp.256-279, 2004.
[6] K. E. Parsopoulos and N. V. Michael, Particle swarm optimization method in multi-objective problems, *Proc. of the 2002 ACM Symposium on Applied Computing*, pp.603-607, 2002.
[7] X. Hu and C. E. Russell, Multi-objective optimization using dynamic neighborhood particle swarm optimization, *Proc. of the World on Congress on COMPUT INTELL*, vol.2, 2002.

[8] X. Hu, C. E. Russell and Y. Shi, Particle swarm with extended memory for multi-objective optimization, *Proc. of the 2003 IEEE Swarm Intelligence Symposium*, 2003.
[9] T. Ray and K. M. Liew, A swarm metaphor for multi-objective design optimization, *Engineering Optimization*, vol.34, no.2, pp.141-153, 2002.
[10] S. Pang, H. Zou, W. Yang et al., An adaptive mutated multi-objective particle swarm optimization with an entropy-based density assessment scheme, *Information & Computational Science*, vol.4, pp.1065-1074, 2013.
[11] M. R. Chen, J. Weng, X. Li et al., Handling multiple objectives with integration of particle swarm optimization and extremal optimization, *Foundations of Intelligent Systems*, pp.287-297, 2014.
[12] V. Roberge, M. Tarbouchi and G. Labonté, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning, *IEEE Transactions on Industrial Informatics*, vol.9, no.1, pp.132-141, 2013.
[13] B. Xue, M. Zhang and W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, *IEEE Transactions on Cybernetics*, vol.43, no.6, pp.1656-1671, 2013.
[14] Y. J. Zheng, H. F. Ling, J. Y. Xue et al., Population classification in fire evacuation: A multiobjective particle swarm optimization approach, *IEEE Transactions on Evolutionary Computation*, vol.18, no.1, pp.70-81, 2014.
[15] D. Izzo, D. Hennes and A. Riccardi, Constraint handling and multi-objective methods for the evolution of interplanetary trajectories, *Journal of Guidance, Control, and Dynamics*, vol.38, no.4, pp.792-800, 2014.
[16] C. Ji, A revised particle swarm optimization approach for multi-objective and multi-constraint optimization, *Proc. of GECCO 2004 Conference*, 2004.
[17] M. J. Reddy and D. Kumar, Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation, *Hydrological Processes*, vol.21, no.21, pp.2897-2909, 2007.
[18] D. Li, X. Li and X. Yu, A multi-objective constraint-handling method with PSO algorithm for constrained engineering optimization problems, *IEEE World Congress on Computational Intelligence*, pp.1528-1535, 2008.
[19] X. Yu and X. Li, A modified PSO algorithm for constrained multi-objective optimization, *The 3rd International Conference on Network and System Security*, pp.462-467, 2009.
[20] C. Worasucheep, Solving constrained engineering optimization problems by the constrained PSO-DD, *The 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, vol.1, pp.5-8, 2008.
[21] G. G. Yen and W. F. Leong, A multiobjective particle swarm optimizer for constrained optimization, *Recent Algorithms and Applications in Swarm Intelligence Research*, 2012.
[22] A. Zhou, B. Y. Qu, H. Li et al., Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation*, vol.1, no.1, pp.32-49, 2011.
[23] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay et al., A survey of multiobjective evolutionary algorithms for data mining: Part I, *IEEE Transactions on Evolutionary Computation*, vol.18, no.1, pp.4-19, 2014.
[24] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, vol.8, no.2, pp.173-195, 2000.
[25] Y. Hajizadeh, M. A. Christie and V. Demyanov, Towards multiobjective history matching: Faster convergence and uncertainty quantification, *SPE Reservoir Simulation Symposium*, 2011.
[26] E. Zitzler and L. Thiele, Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach, *IEEE Trans. Evolutionary Computation*, pp.257-271, 1999.
[27] D. W. Corne, J. D. Knowles and M. J. Oates, The Pareto envelope-based selection algorithm for multi-objective optimization, *Parallel Problem Solving from Nature PPSN VI*, pp.839-848, 2000.
[28] D. W. Corne et al., PESA-II: Region-based selection in evolutionary multi-objective optimization, *Proc. of the Genetic and Evolutionary Computation Conference*, 2001.
[29] D. J. Knowles and D. W. Corne, Approximating the non-dominated front using the Pareto archived evolution strategy, *IEEE Trans. Evolutionary Computation*, pp.149-172, 2000.
[30] E. Zitzler, M. Laumanns and L. Thiele, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, 2001.
[31] M. Erickson, A. Mayer and J. Horn, The niched pareto genetic algorithm 2 applied to the design of groundwater remediation systems, *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, vol.1993, pp.681-695, 2001.

[32] K. Deb et al., A fast and elitist multi-objective genetic algorithm: NSGA-II, *IEEE Trans. Evolutionary Computation*, vol.6, no.2, pp.182-197, 2002.
[33] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, vol.11, no.6, pp.712-731, 2007.
[34] E. Zitzler and S. Künzli, Indicator-based selection in multiobjective search, *Parallel Problem Solving from Nature-PPSN VIII*, pp.832-842, 2004.
[35] N. Beume, B. Naujoks and M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, *European Journal of Operational Research*, vol.181, no.3, pp.1653-1669, 2007.
[36] E. Mezura-Montes and C. A. C. Coello, Constraint-handling in nature-inspired numerical optimization: Past, present and future, *Swarm and Evolutionary Computation*, vol.1, no.4, pp.173-194, 2011.
[37] C. A. C. Coello and A. D. Christiansen, MOSES: A multiobjective optimization tool for engineering design, *Engineering Optimization*, vol.31, no.3, pp.337-368, 1999.
[38] C. A. C. Coello and G. T. Pulido, Multiobjective optimization using a micro-genetic algorithm, *Proc. of the Genetic and Evolutionary Computation Conference*, pp.274-282, 2001.
[39] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, vol.8, no.3, pp.256-279, 2004.
[40] T. Ray and K. S. Won, An evolutionary algorithm for constrained bi-objective optimization using radial slots, *Knowledge-Based Intelligent Information and Engineering Systems*, pp.49-56, 2005.
[41] H. Geng, M. Zhang, L. Huang et al., Infeasible elitists and stochastic ranking selection in constrained evolutionary multi-objective optimization, *Simulated Evolution and Learning*, pp.336-344, 2006.
[42] K. Harada, J. Sakuma, I. Ono et al., Constraint-handling method for multi-objective function optimization: Pareto descent repair operator, *Evolutionary Multi-Criterion Optimization*, pp.156-170, 2007.
[43] B. Y. Qu and P. N. Suganthan, Constrained multi-objective optimization algorithm with an ensemble of constraint handling methods, *Engineering Optimization*, vol.43, no.4, pp.403-416, 2011.
[44] D. A. Van Veldhuizen and G. B. Lamont, Multi-objective evolutionary algorithm research: A history and analysis, *Technical Report TR-98-03*, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.
[45] N. Srinivas and K. Deb, Multi-objective optimization using non-dominated sorting in genetic algorithms, *IEEE Trans. Evolutionary Computation*, vol.2, no.3, pp.221-248, 1994.
[46] Y. G. Woldesenbet, B. G. Tessema and G. G. Yen, Constraint handling in multi-objective evolutionary optimization, *IEEE Trans. Evolutionary Computation*, vol.13, no.3, pp.514-525, 2009.
[47] P. Bao and Q. Zhu, A technique of building non-dominated set based on normalized sort in evolutionary multi-objective optimization, *Acta Electronica Sinica*, vol.9, pp.2010-2015, 2009.
[48] W. Han et al., Comparison study of several kinds of inertia weights for PSO, *IEEE International Conference on PIC*, vol.1, pp.280-284, 2010.
[49] M. A. Abido, Optimal design of power-system stabilizers using particle swarm optimization, *IEEE Trans. Energy Conversion*, vol.17, no.3, pp.406-413, 2002.
[50] M. A. Abido, Optimal power flow using particle swarm optimization, *International Journal of Electrical Power and Energy Systems*, vol.24, no.7, pp.563-571, 2002.
[51] F. Kursawe, A variant of evolution strategies for vector optimization, *Parallel Problem Solving from Nature*, pp.193-197, 1991.
[52] K. Deb, Multi-objective genetic algorithms: Problem difficulties and construction of test problems, *IEEE Trans. Evolutionary Computation*, vol.7, no.3, pp.205-230, 1999.
[53] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, 2001.
[54] M. Tanaka et al., GA-based decision support system for multi-criteria optimization, *IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21st Century*, vol.2, pp.1556-1561, 1995.
[55] G. Chen, M. Xu, Z. M. Wan et al., RLV reentry trajectory multi-objective optimization design based on NSGA-II algorithm, *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, pp.1-6, 2005.
[56] F. Q. Xie, Y. J. Wang, C. Y. Hu et al., A novel migrant PSO algorithm for vehicle reentry trajectory optimization, *Journal of Computational and Theoretical Nanoscience*, vol.9, no.2, pp.276-285, 2012.
[57] X. Jiao and J. Jiang, Multi-objective optimization of reentry trajectory planning for hypersonic aircraft based on ant colony algorithm, *IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp.1203-1208, 2014.
[58] J. Zhao and R. Zhou, Particle swarm optimization applied to hypersonic reentry trajectories, *Chinese Journal of Aeronautics*, 2015.