

RESEARCH ARTICLE

An Improved Multi-Objective Artificial Physics Optimization Algorithm Based on Multi-Strategy Fusion

BAO SUN¹, LIJING ZHANG¹, ZHANLONG LI², KAI FAN¹, QINQIN JIN¹, AND JIN GUO³¹School of Applied Science, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China²School of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China³School of Materials Science and Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi 030024, China

Corresponding author: Zhanlong Li (lizl@tyust.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 52272401 and Grant 51805347, in part by the China Postdoctoral Science Foundation under Grant 2019M661058, and in part by the Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi under Grant 2021L324.

ABSTRACT The elements of the population of the artificial physics optimization (APO) algorithm are assigned mass, velocity, and displacement attributes. It is a new type of heuristic algorithm, but it still has defects such as low efficiency of non-dominated solution selection and unbalanced search capacity of global and local. This paper introduces the mechanism of improved fast non-dominant sorting and partition-guided individual evolution into the APO algorithm to overcome this imperfection. An improved multi-objective artificial physics optimization algorithm based on fast non-dominated sorting (IFNS-MOAPO) is proposed, which is the result of integrating numerous strategies into the APO algorithm. Firstly, an improved fast non-dominated sorting strategy is introduced. This strategy can increase the efficiency of selecting non-dominated solutions and decrease the running time of the algorithm. Secondly, the mechanism of individual evolution guided by partition is proposed. For individuals in infeasible and feasible domains, different mass functions and virtual force calculation rules are adopted to update the algorithm iteratively to boost the convergence performance. To verify the comprehensive performance of the IFNS-MOAPO algorithm, eleven benchmark test problems are selected for simulation experiments and compared with five algorithms in terms of runtime duration, Pareto front plots, and metric values. The results show that the IFNS-MOAPO algorithm has good distribution and can converge to the true Pareto front quickly. It is a useful tool for solving constrained multi-objective optimization problems (CMOPs).

INDEX TERMS Artificial physics optimization, mass function, multi-objective optimization, partition evolution.

I. INTRODUCTION

The single-objective optimization problem (SOP) and the multi-objective optimization problem (MOP) are two branches of optimization problems. The biggest difference between them is the number of objective functions. The objective function of SOP is single, and the optimal solution can be obtained without controversy, so the research results in the optimization field are relatively complete [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Ehab Elsayed Elattar¹.

The sub-objective functions of the MOP are constraints on each other, so it is necessary to make coordination and compromise among all the objective functions to make the overall objective as optimal as possible [3], [4], [5]. The MOP lacks research compared to the SOP. But actually, most problems in engineering applications are MOPs, for example, the optimal path planning problems [6], the distribution network voltage reactive power optimization problems [7], and the production task scheduling problems [8]. There are countless multi-objective optimization algorithms (MOAs), which are effective ways to be used for solving MOPs. The

research of MOAs plays an irreplaceable role both in theory and practice.

To solve MOPs, scholars have done a lot of research on algorithms. From the beginning of the traditional optimization algorithm, and then to the intelligent optimization algorithm, are the condensation of wisdom. Traditional optimization algorithms are generally aimed at structural problems, most of which fall into the category of convex optimization [9]. The essence of the algorithm is to transform multiple objective functions into a single objective function based on some rules and then solve the MOP by a single-objective optimization method [10], [11]. Although the traditional optimization algorithm has a fast convergence speed and a clear termination criterion, it can only find one of the Pareto solution sets of the optimization problem at a time, and the solution results are strongly dependent on the initial values. The emergence of the intelligent algorithm breaks the situation that MOP cannot be solved effectively. And it can better reflect the characteristics of MOP. Well-known intelligent optimization algorithms include evolutionary algorithms (EA) [12] and particle swarm optimization (PSO) algorithms [13]. EA are population-based heuristic search methods that can be solved without prior knowledge.

Due to the various advantages of intelligent algorithms, there has been a recent research boom in the related academic community. As far as multi-objective optimization itself is concerned, representative MOEAs are binary differential evolution with self-learning for multi-objective feature selection [14], a decomposition-based archiving approach for multi-objective evolutionary optimization [15]. These algorithms have high robustness, high nonlinearity, parallelism and wide applicability, and can find globally optimal solutions.

At the same time, MOEAs also have some drawbacks, such as high computational cost, low operational efficiency, and poor diversity of solution sets. To overcome these problems, researchers proposed many improved MOEAs [18], [19]. Meanwhile, other meta-heuristic algorithms are also actively sought to solve MOP. PSO algorithm is one of them. The key points of the study of PSO are as follows: the maintenance of population diversity and distributivity, the selection of non-dominant solutions, and the pruning of the archive set.

The multi-objective particle swarm optimization (MOPSO) algorithm has been widely used to solve MOPs. However, it is prone to fall into the local optimum when solving discrete problems. Therefore, most of the improved algorithms are now based on certain strategies to avoid premature algorithms, and the inclusion of some strategies is accompanied by an increase in algorithm running time [20], [21]. The APO algorithm has some similarities with the PSO algorithm, and its performance can be compared with or even surpasses the classical optimization algorithms [22], [23]. The core of the APO algorithm contains the selection of the non-dominated set, the calculation of the mass function, and the virtual force. Although the RMOAPO algorithm reflects the concept of Pareto domination, it is more complicated to determine the individual number based on individual order

value, neighborhood radius and congestion distance, which increases the computational complexity [24]. The APO algorithm is a new type of algorithm, which is one of the means to solve optimization problems, especially for SOPs, showing powerful solving ability. Based on this, many scholars have tried to apply it to solve MOPs. However, the APO algorithm suffers from the defects of low efficiency of non-dominated solution selection, poor convergence, and distribution in solving MOPs. To address the shortcomings of the traditional APO algorithm with low efficiency of non-dominated solution selection, an improved fast non-dominated sorting strategy is introduced to save the time cost of the algorithm. In the iterative solution of the traditional APO algorithm, it is necessary to first synthesize multiple objectives into one objective, and then find the optimal individual and the worst individual in the population before the final calculation of individual quality, which does not fully reflect the concept of Pareto domination and cannot fully reflect the characteristics of multi-objective optimization [25], [26]. This paper reconstructs the quality function based on individual partitioning and establishes a direct mapping between the calculation of individual quality and individual ordinal values and constraint violation degrees, which fully reflects the characteristics of multi-objective optimization. Also, the mass function and virtual force correction rules distinguish between constrained and unconstrained search according to feasible and infeasible solutions. The simulation results show that the improved algorithm proposed in this paper can accelerate the convergence speed, enhance the diversity of the population, and effectively solve the MOP.

The remainder of this paper is structured as follows. Section 2 introduces the basic definition of constrained multi-objective optimization problems (CMOPs). Section 3 gives a brief introduction to the traditional APO algorithm. Section 4 presents the design of key elements and computational steps of the multi-objective artificial physics optimization algorithm based on improved fast non-dominated sorting (IFNS-MOAPO). In section 5, a comparison experiment with other algorithms is conducted to verify the comprehensive performance of the IFNS-MOAPO algorithm. Finally, Section 6 presents the conclusions.

II. MATHEMATICAL EXPRESSION OF CMOP

The APO algorithm is widely used in practical problems. In these practical problems, optimization models are often constructed with constraints and more than one objective function. In this paper, the mathematical expression of CMOP is defined as:

$$\begin{aligned} \min f(x) &= [f_1(x), f_2(x), \dots, f_k(x)] \\ \text{s.t } p(x) &= (p_1(x), p_2(x), \dots, p_m(x)) < 0 \\ q(x) &= (q_1(x), q_2(x), \dots, q_s(x)) = 0 \\ x &= (x_1, x_2, \dots, x_n) \in R^n, \quad L_i \leq x_i \leq U_i \quad (1) \end{aligned}$$

where k is the number of conflicting objectives, m and s are the respective numbers of inequality constraints and equality

constraints. $x \in R^n$ is the n -dimensional vector of decision variables. U_i and L_i are the upper bound and lower bound of the decision variable $x_i, i = 1, 2, \dots, n$, respectively.

III. APO ALGORITHMS

The APO algorithm is inspired by Newton's force law. Individuals in the population adjust their motion according to the total virtual force and inertia of other individuals to obtain a satisfactory solution. In addition, the individuals in the algorithm are given mass properties, which are constantly updated according to the optimal individual, the worst individual, and their fitness value. As the mass changes, the virtual force of the individual also changes. The rules for calculating the velocity of an individual in the APO algorithm are related to the virtual force and mass of the individual. As the virtual force of the individual changes, the velocity changes accordingly, and thus the position of the individual changes accordingly.

The APO algorithm is described as follows:

Definition 1: The mass function of the individual i .

$$m(i) = e^{\frac{f(x_{best}) - f(x_i)}{f(x_{worst}) - f(x_{best})}} \quad (2)$$

where $f(x_{best})$ denotes the function value of the best individual, $f(x_{worst})$ denotes the function value of the worst individual, and $f(x_i)$ denotes the function value of the individual i .

According to Definition 1, the calculation of individual mass requires the realization of knowing three values. These three values are the fitness value of the best individual, the fitness value of the worst individual, and the fitness value of the individuals in the population, respectively. However, for MOPs, the best and worst individuals cannot be found due to conflicting objective functions.

Definition 2: The virtual force exerted on individual i via individual j in k th dimension.

$$F_{ij,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & f(x_i) > f(x_j) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & f(x_i) \leq f(x_j) \end{cases} \quad \forall i \neq j \text{ and } i \neq best \quad (3)$$

where G is the "gravitational constant" and $x_{j,k} - x_{i,k}$ is the distance from individual i to individual j in the k th dimension.

It can be seen from Definition 2. that the virtual force calculation rules of the traditional APO algorithm do not distinguish individuals in the feasible and infeasible domains. Considering that the test object of this paper is the CMOP, it is necessary to make corrections to the virtual force calculation rules based on individual partitions.

Definition 3: The total virtual force exerted on individual i via all other individuals in the k th dimension.

$$F_{i,k} = \sum_{\substack{j=1 \\ i \neq j}}^n F_{ij,k}, \quad \forall i \neq best \quad (4)$$

Definition 4: The velocity update of the individual i at generation $t + 1$.

$$v_{i,k}(t + 1) = \omega v_{i,k}(t) + \lambda F_{i,k} / m_i, \quad \omega \in (0, 1) \quad (5)$$

where ω is the inertia weight; λ is a random variable generated within $(0, 1)$ with normal distribution.

Definition 5: The position update of the individual i at generation $t + 1$.

$$x_{i,k}(t + 1) = x_{i,k}(t) + v_{i,k}(t + 1) \quad (6)$$

After calculating the total force, velocity, and position of the individual i are updated by using (5) and (6), respectively.

IV. PROPOSED ALGORITHM: IFNS-MOAPO

In this paper, the IFNS-MOAPO algorithm is proposed. Improvements are made based on the APO algorithm as follows: 1) To address the problem of low efficiency of the non-dominated solution selection method of the basic APO algorithm, ameliorated non-dominated sorting strategy is introduced to decrease the running time cost of the algorithm. And each individual is assigned a different ordinal number in combination with the congestion distance. 2) The individual mass function is reconstructed according to the different individual partitions to fully reflect the characteristics of multi-objective optimization. 3) More specific and detailed virtual force calculation rules are constructed according to the different regions of individuals. Through the above improvements, the running efficiency, convergence, and distribution of the algorithm are increased, so that the algorithm will evolve in a better direction, and eventually, the global optimal solution set is found.

A. CONSTRUCTING NON-DOMINATED SETS

The traditional APO algorithm uses Deb's non-dominated sorting method to construct the non-dominated set, which requires all individuals in the population to be compared with each other to generate non-dominated individuals. Therefore, this construction method does not have good structural efficiency.

The improved fast non-dominated sort converts the strategy of comparing one individual with other individuals into a strategy of comparing two individuals with other individuals at the same time, where the two individuals are either unrelated or the second one dominates the first one, and any individual dominated by one of the two individuals is removed, and only the individual not dominated by both individuals is retained. The worst situation of the improved fast non-dominated sort is that the second comparison individual is not found. And then the improved fast non-dominated sort degenerates. But even it is theoretically more efficient than Deb's non-dominated sort.

Let P denote the evolutionary population, x and y are individuals in the population. The definition of individual non-dominant relationships is as follows.

Definition 6: For any individual x and $y, x, y \in P$, we say that x and y are irrelevant if there is no dominant relationship between x and y .

Definition 7: For the certain individual $x \in P$, if there is no $y \in P$ such that $y \succ x$, we say that x is the non-dominant individual of the set P .

Definition 8: We often say $x \succ_d y$ that the relationship \succ_d is not transitive, if $x \succ y$ or x and y are irrelevant.

The IFNS-MOAPO algorithm performs the selection of non-dominated solutions after the population initialization is completed. The selection method is a quicksort algorithm for improved construction of non-dominated sets. After executing the above algorithms, the population reaches the state of stratification based on the level of non-inferior solutions. Then an ordinal number is assigned to each level so that individuals in the same layer have the same non-dominated order. In the following discussion, we use i_{rank} to denote the non-dominated ordinal value of an individual i . For the set F_1 , all individuals are endowed with non-dominated ordinal values $i_{rank} = 1$. For the set F_2 , individuals are assigned the non-dominated ordinal values $i_{rank} = 2$, and so on. Until all individuals in the same stratum in the whole population have the same non-dominated i_{rank} for the entire population. For the individuals in the same stratum, their strengths and weaknesses need to be defined in a certain way. To achieve selective sorting of individuals with the same i_{rank} , crowding and crowding distance comparison operators are introduced. The result is that the individuals in the population can be infinitely close to the true front and uniformly distributed throughout the Pareto domain.

B. PARTITION RECONFIGURATION MASS FUNCTION

The mass function of the traditional MOAPO algorithm is calculated by knowing the objective function values of the best and worst individuals in the population, but the MOP has multiple objective functions, so each sub-objective function needs to be combined into a single objective function through certain strategies. To this end, the mass function is reconstructed in such a way that the mass of individuals in the feasible domain is limited to (1, 2), and the mass function of the individual in the infeasible domain is limited to (0, 1]. The mass function in the IFNS-MOAPO algorithm is modified as:

$$m_i = 1 + e^{-\frac{rank(i)}{N}}, \quad x_i \in \text{feasible} \quad (7)$$

where N is the number of individuals in the population, and $rank(i)$ is a natural number between $1 \sim N$.

$$m_i = e^{-\frac{CV(x_i)}{CV(x)}}, \quad x_i \in \text{infeasible} \quad (8)$$

where $CV(x_i)$ is the constraint violation degree value of individual i and $CV(x)$ is the overall constraint violation.

C. VIRTUAL FORCE CALCULATION RULES FOR PARTITION RECONSTRUCTION

The discussion in this paper is directed to CMOP, which has constraints that limit the range of the optimal solution. The constraints divide the whole search space into feasible design regions and infeasible design regions. The points that satisfy the constraints will fall in the feasible domain, and the points that do not satisfy the constraints will fall in the infeasible domain. The rules for calculating the virtual forces in the APO algorithm should satisfy that individuals with good

adaptation values have a gravitational effect on individuals with poor adaptation values. The rules for the virtual are modified based on the difference in the area where the two individuals are located and are discussed in the following four cases.

$$F_{ij,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & (f(x_i) > f(x_j)) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & (f(x_i) \leq f(x_j)) \end{cases} \quad i, j \in \text{feasible} \quad (9)$$

$$F_{ij,k} = Gm_i m_j (x_{j,k} - x_{i,k}), \quad i \in \text{infeasible}, \quad j \in \text{feasible} \quad (10)$$

$$F_{ij,k} = 0, \quad i \in \text{feasible}, \quad j \in \text{infeasible} \quad (11)$$

$$F_{ij,k} = \begin{cases} Gm_i m_j (x_{j,k} - x_{i,k}), & (CV(x_i) > CV(x_j)) \\ -Gm_i m_j (x_{j,k} - x_{i,k}), & (CV(x_i) \leq CV(x_j)) \end{cases} \quad i, j \in \text{infeasible} \quad (12)$$

When both individuals are in the infeasibility domain, the rule for calculating the virtual force is the magnitude of the constraint violation degree value. The individual with a small constraint violation degree acts as a gravitational force on the individual with a large constraint violation degree, and conversely, the individual with a large constraint violation degree acts as a repulsive force on the individual with a small constraint violation degree.

D. ALGORITHM DESCRIPTION

The effectiveness of MOAPO algorithms lies in the fact that an individual in the population is subject to virtual forces from other individuals and constantly adjusts its position to move in a more optimal direction. The constraint processing technique simply uses the constraint holding method to pull individuals that are not in the feasible domain to the boundary of the feasible domain. It will fall into local search. The constraint violation value is introduced in the improved algorithm to quantify the degree to which a solution violates a constraint and to establish a direct mapping between it and the design of the mass function. The constraint violation $CV(x)$ of a solution x is calculated as:

$$CV(x) = \sum_{m=1}^M \langle P_m(x) \rangle + \sum_{s=1}^S |q_s(x)| \quad (13)$$

where $P_m(x)$ is defined as the m th inequality constraint and $q_s(x)$ is defined as the s th equality constraint.

when x is in the feasible design region, $CV(x) = 0$, a solution x is called a feasible solution. When x is in the infeasible design region, $CV(x) > 0$, x is an infeasible solution.

We use the IFNS-MOAPO algorithm to solve MOPS. Figure 1 shows the design idea of the algorithm.

We use the IFNS-MOAPO algorithm to solve MOPS. The following algorithm is described in pseudocode form.

V. EXPERIMENTAL STUDIES

The ability of the improved algorithm to deal with problems can be judged by the comprehensive performance of the

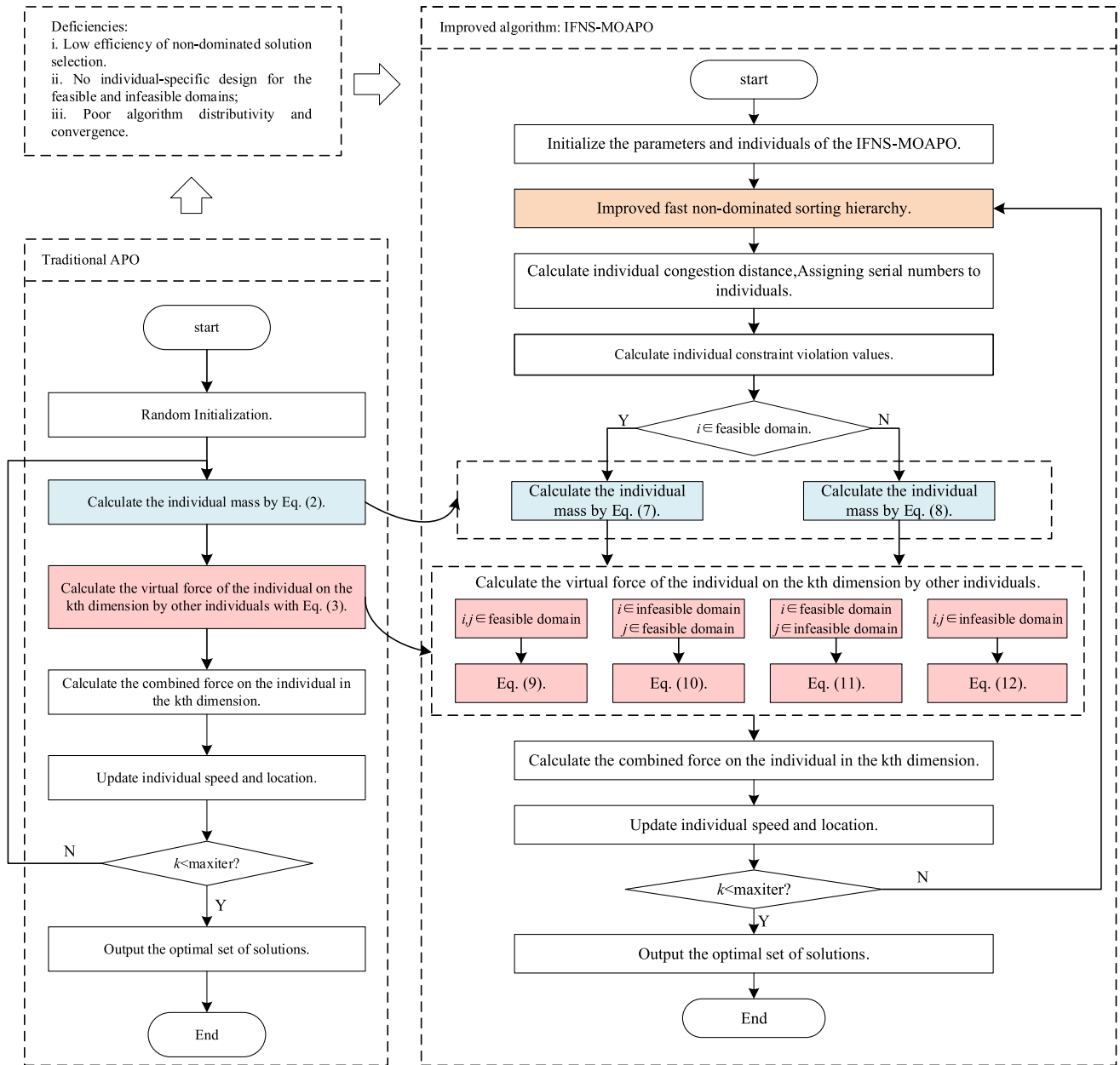


FIGURE 1. The design idea of the algorithm.

improved algorithm on the test function. In the following sections, we first list the names of the test functions. The specific values of the key parameters in the six algorithms are also given. Next, we plot the comparison of the frontier obtained by the six algorithms with the true Pareto. The time complexity of IFNS-MOAPO is analyzed. Then, at the end of this section, we quantitatively analyze the distributivity and convergence of each algorithm by specific metric values.

A. EXPERIMENTAL ENVIRONMENT CONFIGURATION AND BENCHMARK TEST PROBLEMS

The experimental environment of this paper is configured as 11th Gen Intel(R) Core (TM) i5-1135G7 CPU @ 2.50GHz 4.90GHz, 16.00GB, Windows 10 flagship 64-bit OS. The

software environment is MATLAB R2020b (64-bit), and some algorithms implementations rely on the EMO platform.

To verify the effectiveness of IFNS-MOAPO algorithm, the MOPs with constraints are selected as the standard test set [27], [28], as shown in TABLE 1, MOP1-MOP7 are bi-objective test functions, while the Viennet series functions are selected as the standard test set, and the Viennet (VNT) series test functions are tri-objective test functions. The performance of IFNS-MOAPO algorithm and other comparison algorithms are compared through simulation experiments. The comparison algorithms in this paper are NSGA-II [29], CMOPSO [30], MOAPO [25], MOEA/D-DAE [31], CCMO [32], respectively. The reasons for choosing these comparison algorithms are as follows: (i) NSGA-II is an algorithm based

TABLE 1. Test function.

Name	Function description	Constraint conditions	Variable range
Belegundu	$f_1(x, y) = -2x + y; f_2(x, y) = 2x + y$	$-x + y - 1 \leq 0; x + y - 7 \leq 0$	$0 \leq x \leq 5; 0 \leq y \leq 3$
Binh2	$f_1(x, y) = 4x^2 + 4y^2; f_2(x, y) = (x-5)^2 + (y-5)^2$	$(x-5)^2 + y^2 - 25 \leq 0; -(x-8)^2 - (y+3)^2 + 7.7 \leq 0$	$-5 \leq x, y \leq 15$
Srinivas	$f_1(x, y) = (x-2)^2 + (y-1)^2 + 2; f_2(x, y) = 9x - (y-1)^2$	$x^2 + y^2 - 225 \leq 0; x - 3y + 10 \leq 0$	$-20 \leq x, y \leq 20$
Deb	$f_1(x, y) = x; f_2(x, y) = (1+y)/x$	$-9x - y + 6 \leq 0; -9x + y + 1 \leq 0$	$0.1 \leq x \leq 1; 0 \leq y \leq 5$
Osyczka1	$f_1(x, y) = x + y^2; f_2(x, y) = x^2 + y$	$x + y - 12 \leq 0; -x^2 - 10x + y^2 - 16y + 80 \leq 0$	$2 \leq x \leq 7; 5 \leq y \leq 10$
Osyczka2	$f_1(x, y) = -25(x_1 - 2)^2 + (x_2 - 2)^2 + (x_3 - 1)^2 + (x_4 - 4)^2 + (x_5 - 1)^2$ $f_2(x, y) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2$	$-x_1 - x_2 + 2 \leq 0; x_1 + x_2 - 6 \leq 0; -x_1 + x_2 - 2 \leq 0$ $x_1 - 3x_2 - 2 \leq 0; (x_3 - 3)^2 + x_4 - 4 \leq 0; -(x_5 - 3)^2 - x_6 + 4 \leq 0$	$0 \leq x_1, x_2, x_6 \leq 10$ $1 \leq x_3, x_5 \leq 5$ $0 \leq x_4 \leq 6$
Tanaka	$\max F = (f_1(x, y), f_2(x, y))$ $f_1(x, y) = x; f_2(x, y) = y$	$x^2 + y^2 + z^2 \leq 1$	$0 \leq x, y, z \leq 1$

Algorithm 1 IFNS-MOAPO

Input: Population size N_P ; initialize coordinates $x_{i,k}$ and $v_{i,k}$, where $x_{i,k} \in [x_k^{\min}, x_k^{\max}]$ and $v_{i,k} \in [v_k^{\min}, v_k^{\max}]$.
Output: Pareto front found by the IFNS-MOAPO algorithm.

- 1: **procedure** IFNS-MOAPO
- 2: **Initialize population**
- 3: **Pareto Front (PF)**
- 4: Improved fast non-dominated sort stratification
- 5: Calculate the individual's crowding distance
- 6: Assign each individual an ordinal number $rank(i)$
- 7: **while** ($t < Max_{iterations}$) **do**
- 8: **for** each individual from the archive **do**
- 9: calculate the constraint violation value for each individual by (13).
- 10: **if** $i \in feasible\ domain$, **then**
- 11: calculate the individual's mass with (7).
- 12: **else**
- 13: calculate the individual's mass with (8).
- 14: **end if**
- 15: select the corresponding virtual force calculation rules to calculate the force receives from other individuals in the k th dimension by (9)-(12).
- 16: calculation of the combined virtual forces by (4).
- 17: update the speed and position of each individual by (5) and (6).
- 18: **end do**
- 19: $t = t + 1$
- 20: update Pareto front by the algorithm.
- 21: **end while**
- 22: **return Pareto Front (PF)**
- 23: **end procedure**

on non-dominated sorting, and it is used as a comparison to test the effectiveness of the improved non-dominated sorting method in the IFNS-MOAPO algorithm to save time costs. (ii) The reason for choosing MOPSO algorithm as the comparison object is that the essence of traditional APO

TABLE 2. Parameter description.

Algorithm name	Parameter setting
NSGA-II	$P_m = 0.01; P_c = 0.8$
CMOPSO	$R_1, R_2 \in [0, 1]$
MOAPO	$w_i = 0.9; w_f = 0.4; G = 10$
MOEA/D-DAE	$\alpha = 0.95$
CCMO	$\tau = 0.1; cp = 2; l = 20$
IFNS-MOAPO	$G_i = 100; G_f = 1; \omega \in (0, 1)$

algorithm is similar to that of particle swarm algorithm. (iii) Compare with the traditional MOAPO algorithm and analyze the effectiveness of the improved strategy in the IFNS-MOAPO algorithm. (iv) MOEA/D-DAE and CCMO have excellent performance and are relatively advanced algorithms in recent years. Compared with these two algorithms, the comprehensive performance of the improved algorithm can be better evaluated. The other parameters for each compared algorithm are set according to the original paper. The specific values of the key parameters in the six algorithms are shown in TABLE 2.

B. EXPERIMENTAL RESULTS OF THE FINAL NON-DOMINATED FRONT

The comparison between the Pareto front obtained by the algorithm and the true front can directly reflect the degree of approximation to the real front and the advantages and disadvantages of the distribution of the algorithm. Figure 2 shows the comparison plot of the Pareto front obtained by the six algorithms with the true Pareto front. For each test function, there are six plots: the one on the right is for IFNS-MOAPO, and the others are for NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, and CCMO.

As Figure 2 shows, the improved algorithm proposed in this paper and the other six algorithms can find the non-inferior solution set of the test function. The improved algorithm in this paper can not only find feasible individuals but also these individuals are very close to or even coincide with the true Pareto frontier, which meets our expectations.

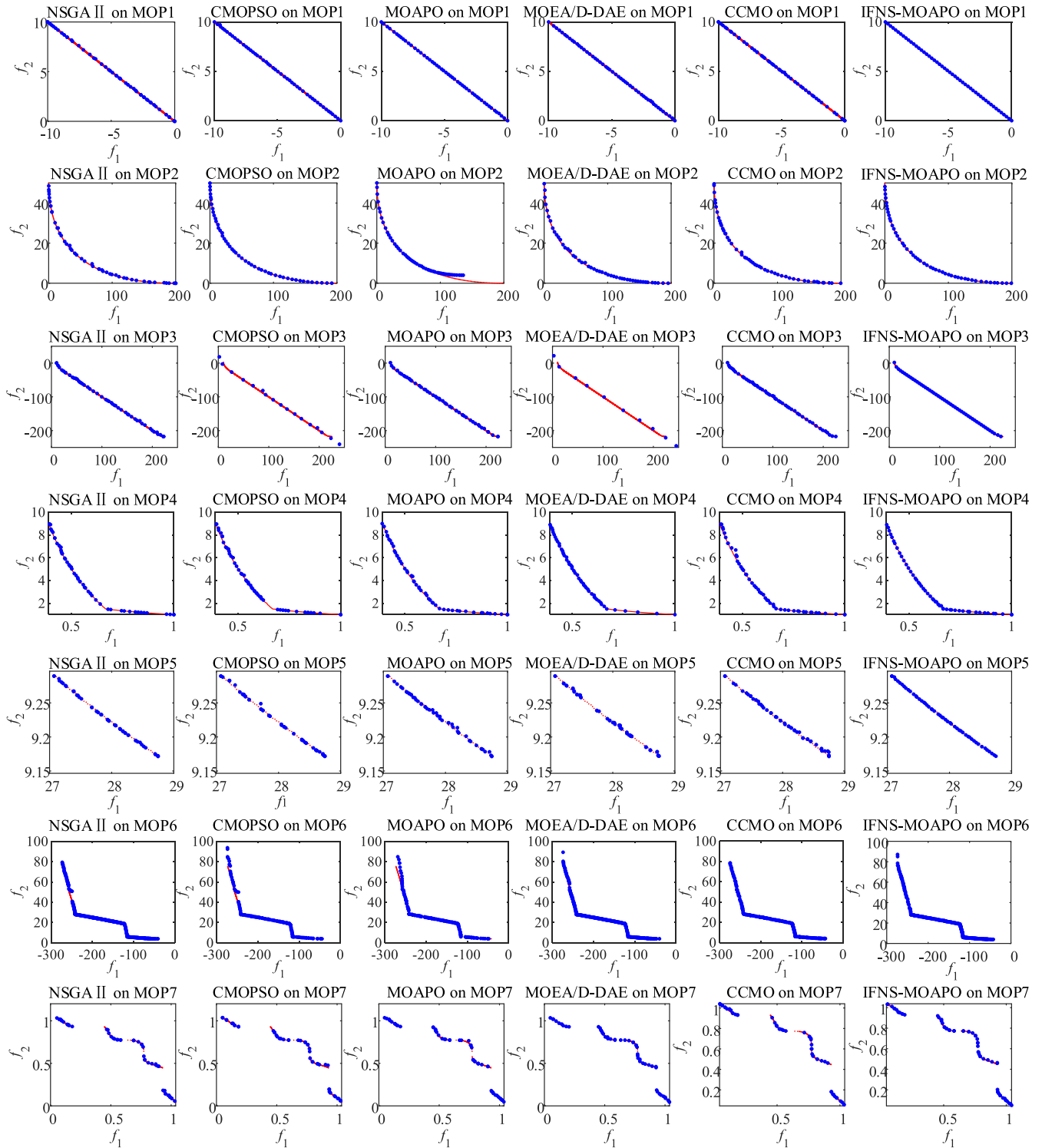


FIGURE 2. The comparison plot of the Pareto front is obtained by the six algorithms with the true Pareto front.

C. ALGORITHM EFFICIENCY

The algorithm efficiency can be analyzed from two aspects. From the theoretical aspect, we can analyze the algorithm’s time complexity. From the application aspect, we can evaluate the running time taken by the algorithm. In this paper, the time

complexity of non-dominated solution selection is changed from $O(kN^2)$ to $O(kN \log N)$ after the improvement of the traditional MOAPO algorithm [33, 34]. TABLE 3 shows the average computation time of the algorithm for 30 runs on the seven test functions.

TABLE 3. Comparison of the running time of six algorithms on 7 test functions. (NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, CCMO, and IFNS-MOAPO).

Problem	NSGAI	CMOPSO	MOAPO	MOEA/D-DAE	CCMO	IFNS-MOAPO
MOP1	2.37s	3.26s	13.25s	37.92s	11.76s	1.82s
MOP2	1.92s	4.72s	11.49s	37.50s	11.60s	2.52s
MOP3	2.05s	12.64s	11.57s	48.86s	17.60s	1.55s
MOP4	1.86s	6.62s	11.73s	44.47s	15.57s	1.49s
MOP5	1.99s	4.71s	12.08s	34.29s	15.10s	2.32s
MOP6	1.86s	12.37s	11.65s	43.62s	16.96s	1.57s
MOP7	1.89s	3.97s	11.91s	40.01s	15.48s	1.59s

The algorithm runtime result statistics in Table 3 show that IFNS-MOAPO takes the shortest time on most of the tested functions and MOEA/D-D-DAE takes the longest time. Although the MOEA/D-D-DAE and CCMO achieved good results for both HV and IGD statistics on some MOP test functions, the MOEA/D-D-DAE and CCMO were time-consuming compared to IFNS-MOAPO, especially MOEA/D-D-DAE. Compared with the MOAPO algorithm, the running time of IFNS-MOAPO on the MOP test function is significantly reduced, which indicates that the improved fast non-dominated sorting strategy in this paper is effective. The introduction of the improved fast non-dominated sorting strategy not only makes the algorithm time complexity decrease in the theoretical aspect but also makes the running efficiency of IFNS-MOAPO improve in the practical application.

D. PERFORMANCE METRICS

The convergence and distribution of the improved algorithm can be illustrated by scalar values. The common performance evaluation indexes include GD, SP, HV, and IGD. HV and IGD are comprehensive performance evaluation indexes. The calculation of the HV metric strictly follows the Pareto dominance principle, the larger the HV value, the better the algorithm performance. the IGD metric is the reverse mapping of the GD metric. the smaller the value of the IGD metric, the better the performance of the algorithm. The calculation of the HV index value does not need to know the Pareto optimal

surface. HV and IGD are defined as:

$$HV = \lambda(U_{i=1}^{|S|} v_i) \tag{14}$$

$$IGD = \left(\sum_{\bar{j} \in PF^*} \min_{\bar{i} \in P} |\bar{j} - \bar{i}| \right) / n \tag{15}$$

where S is the nondominant set, v_i denotes the supervolume of reference points and non-dominant individuals, λ is the Lebesgue measure. n is the number of vectors in the frontier after running the algorithm. $\min_{\bar{i} \in P} |\bar{j} - \bar{i}|$ denotes the minimum Euclidean distance from the point \bar{j} on the Pareto optimal surface to the individual \bar{i} in the optimal solution set P .

E. EXPERIMENTAL RESULTS ON TEST PROBLEMS

The IFNS-MOAPO algorithm is a random search algorithm. Due to the randomness of the algorithm, the results of a single experiment are not representative. Therefore, to avoid the errors caused by the random search of the algorithm, TABLE 4 and TABLE 5 show the comparison results of the mean and standard deviation (mean vs. std) of HV metrics and IGD metrics between the IFNS-MOAPO algorithm and NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, CCMO after 30 consecutive independent runs on MOP1-7, respectively, where the best results are shown in bold.

TABLE 4 shows that the improved algorithm IFNS-MOAPO algorithm performs well on the MOP1, MOP2, MOP3, MOP4, MOP5, and MOP7, outperforming the

TABLE 4. Comparison of the mean and standard deviation of HV metrics on MOP test functions for six algorithms.

Problem	NSGAI	CMOPSO	MOAPO	MOEADDAE	CCMO	IFNS-MOAPO
MOP1	4.6062e-1(1.06e-2) -	2.0234e-1(1.13e-1) -	3.9622e-1(1.96e-2) -	5.2270e-1(8.14e-3) -	5.2001e-1(5.09e-3) -	5.4047e-1(6.58e-3)
MOP2	5.4010e-1(3.98e-2) -	7.0570e-1(2.84e-3) -	4.7662e-1(3.28e-2) -	7.0809e-1(4.35e-3) -	7.0263e-1(4.58e-3) -	7.1732e-1(7.50e-4)
MOP3	5.5040e-1(2.95e-2) -	3.8275e-1(2.19e-2) -	5.3388e-1(2.37e-2) -	2.3323e-1(1.55e-2) -	5.7966e-1(2.82e-2) -	6.2593e-1(2.54e-2)
MOP4	5.3504e-1(3.11e-2) -	5.8061e-1(3.02e-2) -	5.2256e-1(8.00e-2) -	5.4529e-1(3.84e-2) -	5.3658e-1(3.29e-2) -	6.4141e-1(1.07e-2)
MOP5	3.9699e-1(7.50e-2) -	4.6627e-1(9.66e-2) -	5.3536e-1(2.34e-2) -	3.6396e-1(2.97e-2) -	5.1055e-1(9.53e-3) -	5.5374e-1(1.13e-2)
MOP6	5.1644e-1(3.48e-2) =	5.2865e-1(2.50e-3) -	5.1312e-1(3.70e-3) -	5.3603e-1(3.72e-3) +	5.3737e-1(2.06e-3) +	5.3392e-1(2.63e-3)
MOP7	3.6218e-1 (6.90e-2) -	3.3806e-1(1.83e-3) -	3.7924e-1(5.04e-3) -	4.0625e-1(1.80e-3) -	4.0481e-1(2.01e-3) -	4.0799e-1(1.23e-3)
+/-/=	0/6/1	0/7/0	0/7/0	1/6/0	1/6/0	

TABLE 5. Comparison of the mean and standard deviation of IGD metrics on MOP test functions for six algorithms.

Problem	NSGAII	CMOPSO	MOAPO	MOEADDAE	CCMO	IFNS-MOAPO
MOP1	8.2378e-2(7.89e-3) -	5.1095e-1(2.01e-1) -	1.2970e-1(1.61e-2) -	3.4828e-2(6.71e-3) -	3.8490e-2(4.44e-3) -	2.2674e-2(5.85e-3)
MOP2	1.3694e-1(3.14e-2) -	1.2845e-2(2.04e-3) -	1.6791e-1(2.55e-2) -	1.1442e-2(4.32e-3) -	1.5252e-2(3.40e-3) -	5.0530e-3(4.06e-4)
MOP3	8.7471e-2(2.42e-2) -	3.8275e-1(2.19e-2) -	8.8074e-2(1.38e-2) -	2.3323e-1(1.55e-2) -	6.9508e-2(2.02e-2) -	3.7955e-2(1.81e-2)
MOP4	2.4069e-1(5.73e-2) -	6.7456e-2(2.85e-2) =	1.3705e-1(5.42e-2) -	1.0634e-1(5.80e-2) -	2.3130e-1(6.94e-2) -	7.1972e-2(1.81e-2)
MOP5	1.4111e-1(6.42e-2) -	8.6289e-2(8.60e-2) -	3.5251e-2(2.04e-2) -	8.7668e-2(2.77e-2) -	4.7049e-2(5.88e-3) -	2.0516e-2(6.89e-3)
MOP6	3.4159e-2(4.77e-2) =	5.0265e-2(1.01e-3) -	2.5391e-2(1.88e-3) -	1.0201e-2(2.24e-3) +	9.5227e-3(1.40e-3) +	1.3720e-2(2.28e-3)
MOP7	1.2537e-1(1.86e-1) -	2.0151e-1(3.18e-3) -	2.5407e-2(2.91e-3) -	1.2745e-2(2.04e-3) -	1.1982e-2(2.10e-3) -	1.0669e-2(3.05e-3)
+/-/=	0/6/1	0/6/1	0/7/0	1/6/0	1/6/0	

TABLE 6. Comparison of the mean and standard deviation of HV metrics on VNT test functions for six algorithms.

Problem	NSGAII	CMOPSO	MOAPO	MOEADDAE	CCMO	IFNS-MOAPO
VNT1	3.4119e-1 (6.98e-4) -	3.4407e-1 (2.56e-4) -	3.4013e-1 (9.18e-4) -	3.4372e-1 (4.06e-4) -	3.3371e-1 (1.70e-3) -	3.4446e-1 (3.24e-4)
VNT2	3.3142e-1 (3.35e-4) -	3.3301e-1 (9.92e-5) =	3.3326e-1 (3.06e-3) =	3.3305e-1 (6.94e-5) +	3.3355e-1 (2.17e-3) +	3.3297e-1 (1.07e-4)
VNT3	1.7581e-1 (5.20e-5) +	1.7582e-1 (4.08e-5) +	1.7389e-1 (5.68e-4) -	1.7595e-1 (2.85e-5) +	1.7335e-1 (6.62e-4) -	1.7513e-1 (8.97e-5)
VNT4	2.5753e-1 (4.25e-4) -	2.3369e-1 (7.02e-3) -	2.5657e-1 (6.53e-4) -	2.5829e-1 (2.87e-4) -	2.5022e-1 (1.50e-3) -	2.5905e-1 (1.25e-4)
+/-/=	1/3/0	1/2/1	0/3/1	2/2/0	1/3/0	

TABLE 7. Comparison of the mean and standard deviation of IGD metrics on VNT test functions for six algorithms.

Problem	NSGAII	CMOPSO	MOAPO	MOEADDAE	CCMO	IFNS-MOAPO
VNT1	1.5751e-1 (6.22e-3) -	1.3071e-1 (2.31e-3) -	1.6789e-1 (7.44e-3) -	1.3182e-1 (3.14e-3) -	2.1634e-1 (8.62e-3) -	1.2648e-1 (3.29e-3)
VNT2	2.2146e-2 (1.46e-3) -	1.2898e-2 (3.76e-4) =	5.3068e-2 (1.84e-2) -	1.3702e-2 (5.00e-4) -	7.3430e-2 (1.18e-2) -	1.2733e-2 (4.28e-4)
VNT3	4.4652e-2 (1.25e-2) -	3.2162e-2 (1.66e-3) -	1.7512e+0 (2.60e-1) -	4.9718e-1 (4.57e-1) -	2.1606e-1 (6.69e-2) -	3.1260e-2 (1.37e-3)
VNT4	1.5492e-1 (8.78e-3) -	1.1798e+0 (1.19e-1) -	2.5209e-1 (6.68e-2) -	1.4357e-1 (1.38e-2) -	3.9032e-1 (9.47e-2) -	1.1529e-1 (8.01e-3)
+/-/=	0/4/0	0/3/1	0/4/0	0/4/0	0/4/0	

comparison algorithms NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, and CCMO. The MOEA/D-DAE and CCMO outperform the IFNS-MOAPO algorithm on the test function MOP6. In terms of significant differences, the NSGA-II does not differ significantly from the IFNS-MOAPO algorithm on MOP6.

TABLE 5 shows that the improved algorithm IFNS-MOAPO algorithm has lower mean values of IGD metrics on the five test functions MOP1, MOP2, MOP3, MOP5, and MOP7 than the NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE and CCMO, and the distribution of the obtained test results is better. In terms of significant differences, the CCMO outperforms the IFNS-MOAPO algorithm on the test function MOP6.

TABLE 6 TABLE 4 and TABLE 7 show the comparison results of the mean and standard deviation (mean vs. std) of HV metrics and IGD metrics between the IFNS-MOAPO algorithm and NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, CCMO after 30 consecutive independent runs on VNT series test functions, respectively, where the best results are shown in bold.

TABLE 6 shows that the improved algorithm performs well on VNT1 and VNT4 compared to the comparison algorithm.

TABLE 7 shows that the mean values of IGD metrics of the IFNS-MOAPO algorithm on the VNT series test functions are lower than those of the comparison algorithm, which fully illustrates that the IFNS-MOAPO algorithm also has good distributivity and convergence on the three-objective test functions.

The conventional MOAPO algorithm has the problem of partitioning ambiguity in the calculation rules of mass function and virtual force, which leads to the inferiority of HV and IGD index values to the IFNS-MOAPO algorithm on most of the test functions selected in this paper; the CMOPSO also has the inferior performance to the IFNS-MOAPO algorithm on the test functions due to its tendency to fall into local optimum.

The box plots can visually reflect the dispersion of the sample data as well as the symmetry and tail weight of the overall distribution. Figure 3 and Figure 4 show the box plots of HV indicators and IGD indicators for the six optimization algorithms on the test functions, where the horizontal coordinates label 1, 2, 3, 4, 5, and 6 correspond to the algorithms as follows: 1-NSGA-II, 2-CMOPSO, 3-MOAPO, 4-MOEA/D-DAE, 5-CCMO, and 6-IFNS-MOAPO.

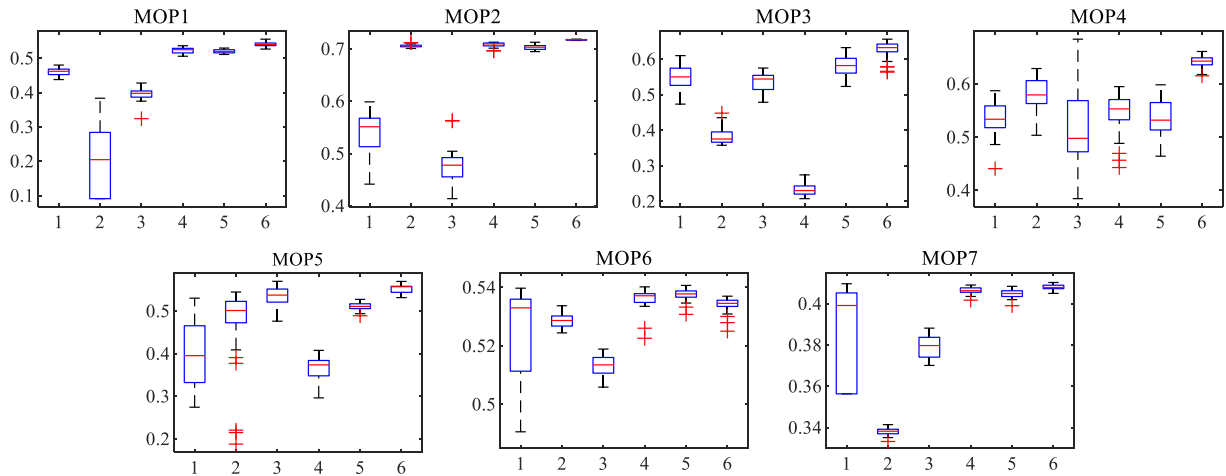


FIGURE 3. Box plots of HV value on test functions from the six algorithms in 30 independent runs. 1, 2, 3, 4, 5, and 6 represent NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, CCMO, and IFNS-MOAPO.

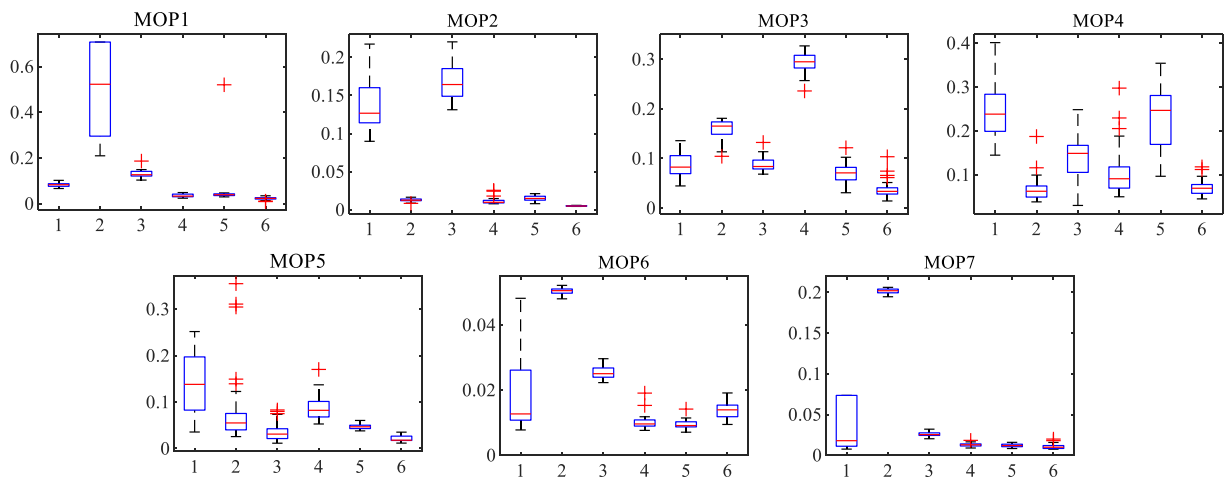


FIGURE 4. Box plots of IGD value on test functions from the six algorithms in 30 independent runs. 1, 2, 3, 4, 5, and 6 represent NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, CCMO, and IFNS-MOAPO.

From Figure 3, we can observe that IFNS-MOAPO has superiority for the HV statistic followed by CCMO. For instance, IFNS-MOAPO shows higher HV statistical results on most of the seven MOP test problems. Compared with the MOAPO algorithm, the HV statistics of IFNS-MOAPO are significantly improved, indicating that the strategy of partitioning to reconstruct the mass function and virtual force rules is effective.

From Figure 4, we can observe that IFNS-MOAPO has superiority for the IGD statistic followed by CCMO and MOEA/D-DAE. The CMOPSO algorithm shows high IGD statistics on MOP1, MOP6, and MOP7 test problems with poor distribution. The effectiveness of the improved strategy in this paper is further illustrated by comparing the IGD statistics results with those of the traditional MOAPO algorithm. Establishing a direct mapping of the mass function to the individual ordinal number and reconstructing the virtual force calculation rules according to the region can significantly improve the distributivity and convergence of the IFNS-MOAPO algorithm.

F. APPLICATION ON REAL-WORLD ENGINEERING OPTIMIZATION

The disc brake design problem is used to further test the effectiveness of the algorithm in this paper [35]. The problem has two objectives, which are the weight of the brake and the stopping time. The problem contains four variables, x_1 and x_2 denote the diameters of the inner and outer rings of the brake, x_3 denotes the engagement force, and x_4 denotes the number of friction surfaces. The constraints of the problem are the minimum distance between radii, the maximum length of the brake, pressure, temperature, and torque limits, respectively. The specific optimization problem and constraints are shown in Eq. (16).

$$\begin{aligned}
 & \text{Minimize } f_1(x) = 4.9 \times 10^{-5}(x_2^2 - x_1^2)(x_4 - 1) \\
 & \text{Minimize } f_2(x) = 9.82 \times 10^6(x_2^2 - x_1^2)/(x_3x_4(x_2^3 - x_1^3)) \\
 & \text{Subject to } g_1(x) = (x_2 - x_1) - 20 \geq 0 \\
 & \quad g_2(x) = 30 - 2.5(x_4 + 1) \geq 0 \\
 & \quad g_3(x) = 0.4 - x_3/(3.14(x_2^2 - x_1^2)) \geq 0
 \end{aligned}$$

TABLE 8. Mean and standard deviation of HV results for NSGAII, CMOPSO, MOAPO, MOEADDAE, CCMO, and IFNS-MOAPO on real-world engineering optimization problem.

HV	NSGAII	CMOPSO	MOAPO	MOEADDAE	CCMO	IFNS-MOAPO
mean	8.5739E-01	NaN	8.4959E-01	8.5327E-01	8.6036E-01	8.6442E-01
std	2.00E-02	NaN	2.06E-02	1.90E-02	1.83E-02	1.73E-02
+/-/=	-	-	-	-	-	-

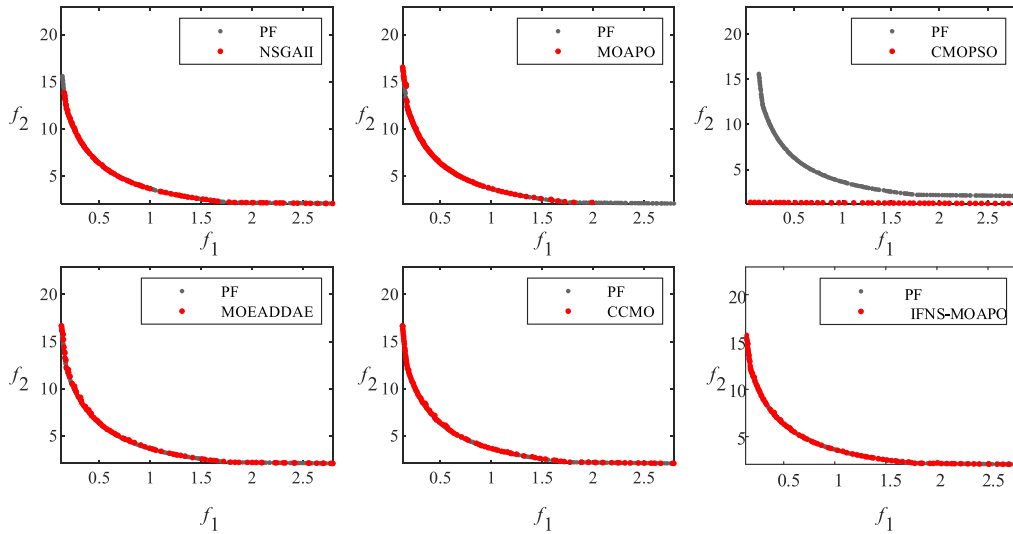


FIGURE 5. Pareto front obtained by the six algorithms on disk brake problem.

$$\begin{aligned}
 g_4(x) &= 1 - 2.22 \times 10^{-3} x_3(x_2^3 - x_1^3) / \\
 &(x_2^2 - x_1^2)^3 \geq 0 \\
 g_5(x) &= 2.66 \times 10^{-2} x_3 x_4 (x_2^3 - x_1^3) / \\
 &(x_2^2 - x_1^2) \geq 0
 \end{aligned} \tag{16}$$

where $55 \leq x_1 \leq 80$, $75 \leq x_2 \leq 110$, $1000 \leq x_3 \leq 3000$, and $2 \leq x_4 \leq 20$.

Each algorithm was run 30 times independently on this problem, with the number of function evaluations set to 100000 and population size of 100. We used HV to evaluate the performance of the IFNS-MOAPO, and the results after 30 runs are summarized in TABLE 8. The Pareto front plots obtained for the six algorithms are given in Figure 5.

As TABLE 8 shows, the result of IFNS-MOAPO is significantly better than the other five algorithms on this real-world problem in terms of HV metric. CCMO achieves the second-best result followed by NSGAII and MOEADDAE. CMOPSO and MOAPO obtain the worst results.

As Figure 5 shows, NSGA-II, MOAPO, MOEADDAE, and CCMO demonstrate worse performance on distribution compared to IFNS-MOAPO because the individual distribution at the ends of their approximate PFs is sparse. All points of CMOPSO cannot converge to the PF, so CMOPSO is not valid for this practical engineering problem. Therefore, IFNS-MOAPO achieves the best performance concerning convergence and distribution compared to the other peer algorithms.

VI. CONCLUSION AND FUTURE STUDIES

In this paper, we propose a new multi-objective optimization algorithm, namely IFNS-MOAPO, for the CMOPs. The improving fast non-dominated sorting strategy, the conception of partition reconstruction mass function, and the conception of partition reconstruction virtual force calculation rules have made the IFNS-MOAPO algorithm more effective in solving CMOPs. By comparing IFNS-MOAPO with other optimization algorithms (NSGA-II, CMOPSO, MOAPO, MOEA/D-DAE, and CCMO), we demonstrate that our IFNS-MOAPO is efficient in dealing with CMOPs. The main findings of the paper can be summarized as follows.

- (1) On the basis of the traditional APO algorithm, the improved fast non-dominated ranking strategy can efficiently reduce the run time of our algorithm. For most of the test functions, the IFNS-MOAPO algorithm has the fastest running time, although this advantage is very small.
- (2) The mass function of partition reconstruction and the virtual force calculation method make full use of the characteristics of the feasible and infeasible domains in the CMOPs. The improved strategy speeds up the convergence of the algorithm and enhances the diversity of populations.

The IFNS-MOAPO algorithm is strongly influenced by the parameters, especially the problem of adjusting the gravitational factor and inertial weights. In our future work, we will

investigate the parameter setting problem and the computational complexity of the algorithm. In addition, this paper is only a theoretical analysis of the IFNS-MOAPO algorithm. In the future, we will use the IFNS-MOAPO algorithm to solve MOPs in practical engineering, such as engineering vehicle suspension parameters, robot path planning, and logistics allocation.

REFERENCES

- [1] E. Segredo, C. Segura, C. León, and E. Hart, "A fuzzy logic controller applied to a diversity-based multi-objective evolutionary algorithm for single-objective optimisation," *Soft Comput.*, vol. 19, no. 10, pp. 2927–2945, Sep. 2014.
- [2] H. Liu, W. Du, and Z. Guo, "A multi-population evolutionary algorithm with single-objective guide for many-objective optimization," *Inf. Sci.*, vol. 503, pp. 39–60, Nov. 2019.
- [3] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017.
- [4] S. Santander-Jimenez, M. A. Vega-Rodriguez, and L. Sousa, "Multiobjective frog-leaping optimization for the study of ancestral relationships in protein data," *IEEE Trans. Evol. Comput.*, vol. 22, no. 6, pp. 879–893, Dec. 2018.
- [5] G. Dhiman and V. Kumar, "Multi-objective spotted hyena optimizer: A multi-objective optimization algorithm for engineering problems," *Knowl.-Based Syst.*, vol. 150, pp. 175–197, Jun. 2018.
- [6] E. Barnett and C. Gosselin, "A bisection algorithm for time-optimal trajectory planning along fully specified paths," *IEEE Trans. Robot.*, vol. 37, no. 1, pp. 131–145, Feb. 2021.
- [7] L. Chen, Z. Deng, and X. Xu, "Two-stage dynamic reactive power dispatch strategy in distribution network considering the reactive power regulation of distributed generations," *IEEE Trans. Power Syst.*, vol. 34, no. 2, pp. 1021–1032, Mar. 2019.
- [8] Q. Qi, L. Zhang, J. Wang, H. Sun, Z. Zhuang, J. Liao, and F. R. Yu, "Scalable parallel task scheduling for autonomous driving using multi-task deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13861–13874, Nov. 2020.
- [9] M. Maalouf, D. Homouz, and T. B. Trafails, "Logistic regression in large rare events and imbalanced data: A performance comparison of prior correction and weighting methods," *Comput. Intell.*, vol. 34, no. 1, pp. 161–174, Aug. 2017.
- [10] B. Pérez-Cañedo, J. L. Verdegay, and R. M. Pérez, "An epsilon-constraint method for fully fuzzy multiobjective linear programming," *Int. J. Intell. Syst.*, vol. 35, no. 4, pp. 600–624, Apr. 2020.
- [11] H. Garg, "A linear programming method based on an improved score function for interval-valued Pythagorean fuzzy numbers and its application to decision-making," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 26, no. 1, pp. 67–80, 2018.
- [12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [13] Y. Hu, Y. Zhang, and D. Gong, "Multiobjective particle swarm optimization for feature selection with fuzzy cost," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 874–888, Feb. 2021.
- [14] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
- [15] Y. Zhang, D. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Inf. Sci.*, vols. 430–431, pp. 397–413, Mar. 2018.
- [16] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, Sep. 2007.
- [17] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011.
- [18] Y. Wang, Z. Dong, T. Hu, and X. Wang, "An improved MOEA/D algorithm for the carbon black production line static and dynamic multiobjective scheduling problem," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2020, pp. 1–8.
- [19] W.-X. Wang, K.-S. Li, X.-Z. Tao, and F.-H. Gu, "An improved MOEA/D algorithm with an adaptive evolutionary strategy," *Inf. Sci.*, vol. 539, pp. 1–15, Oct. 2020.
- [20] N. Al Moubayed, A. Petrovski, and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, Mar. 2014.
- [21] R. Nshimirimana, A. Abraham, and G. Nothnagel, "A multi-objective particle swarm for constraint and unconstrained problems," *Neural Comput. Appl.*, pp. 11355–11385, Jan. 2021.
- [22] A. Gorbenko and V. Popov, "The force law design of artificial physics optimization for robot anticipation of motion," *Adv. Stud. Theor. Phys.*, vol. 6, no. 13, pp. 625–628, 2012.
- [23] X. Zhan, T. Xiang, H. Chen, B. Zhou, and Z. Yang, "Vulnerability assessment and reconfiguration of microgrid through search vector artificial physics optimization algorithm," *Int. J. Electr. Power Energy Syst.*, vol. 62, pp. 679–688, Nov. 2014.
- [24] Y. Wang and J.-C. Zeng, "A multi-objective artificial physics optimization algorithm based on ranks of individuals," *Soft Comput.*, vol. 17, no. 6, pp. 939–952, Feb. 2013.
- [25] L. Xie, J. Zeng, and Q. Yang, "A predictable artificial physics optimisation algorithm," *Int. J. Comput. Sci. Math.*, vol. 6, no. 5, pp. 459–470, 2015.
- [26] K. Teeparthi and D. M. V. Kumar, "Multi-objective hybrid PSO-APO algorithm based security constrained optimal power flow with wind and thermal generators," *Eng. Sci. Technol. Int. J.*, vol. 20, pp. 411–426, Apr. 2017.
- [27] T. Ray, K. Tai, and K. C. Seow, "An evolutionary algorithm for multiobjective optimization," *Eng. Optim.*, vol. 33, no. 3, pp. 299–424, 2001.
- [28] K. Deb, "Multi-objective optimization using evolutionary algorithms: An introduction," *Comput. Optim. Appl.*, vol. 39, no. 1, pp. 75–96, Feb. 2017.
- [29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Jan. 2002.
- [30] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Inf. Sci.*, vol. 427, pp. 63–76, Feb. 2018.
- [31] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 938–947, Oct. 2020.
- [32] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, "A coevolutionary framework for constrained multi-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 102–116, Mar. 2021.
- [33] S. Abasabadi, H. Nematzadeh, H. Motameni, and E. Akbari, "Automatic ensemble feature selection using fast non-dominated sorting," *Inf. Syst.*, vol. 100, Sep. 2021, Art. no. 101760.
- [34] M. S. Hossain, S. Mondal, R. S. Ali, and M. Hasan, "Optimizing complexity of quick sort," in *Computing Science, Communication and Security*. Berlin, Germany: Springer, 2020, pp. 329–339.
- [35] T. Ray and K. M. Liew, "A swarm metaphor for multiobjective design optimization," *Eng. Optim.*, vol. 34, no. 2, pp. 141–153, Oct. 2002.



BAO SUN was born in Taigu, Jinzhong, Shanxi, China, in 1981. He received the B.S. degree in mathematics from Taiyuan Normal University, in 2004, and the M.S. degree in mathematics and the Ph.D. degree in mechanical engineering from the Taiyuan University of Science and Technology, Shanxi, China, in 2009 and 2015, respectively. He is currently an Associate Professor with the College of Applied Science, Taiyuan University of Science and Technology. His research interests include theoretical research on intelligent optimization algorithms, vibration and noise reduction for construction vehicles, and applications of optimization algorithms in engineering.



LI JING ZHANG was born in Xiaoyi, Shanxi, China in 1997. She received the B.S. degree in mathematics from Shanxi Datong University, in 2019. She is currently pursuing the M.S. degree with the Department of Mathematics, College of Applied Science, Taiyuan University of Science and Technology, China. Her research interests include improvement of intelligent optimization algorithms, vibration and noise control of engineering vehicles, and optimization design of vehicle performance.



ZHANLONG LI was born in Shuozhou, Shanxi, China, in 1985. He received the B.S. degree in material processing and control engineering from Xi'an Polytechnic University, in 2010, the M.S. degree in mechanical design and theory from the Taiyuan University of Science and Technology, in 2013, and the Ph.D. degree in vehicle engineering from the Xi'an University of Technology, Shaanxi, China, in 2016. He is currently an Associate Professor with the College of Mechanical

Engineering, Taiyuan University of Science and Technology. He is also working as the Deputy Director with the Construction Machinery and Vehicle Research Institute. He is mainly responsible for the daily management at the Shanxi New Energy Vehicle Engineering Technology Center and Academician workstation. His research interests include special vehicle dynamics and control, new energy drive technologies, and the integration of optimization algorithms with engineering practice.



KAI FAN was born in Zigong, Sichuan, China, in 1990. He received the B.S. degree in mathematics and applied mathematics and the M.S. degree in fundamental mathematics from Northeastern University, in 2012 and 2014, respectively, and the Ph.D. degree in mechanical engineering from the Taiyuan University of Science and Technology, Shanxi, China, in 2021. He is currently a Lecturer with the Department of Mathematics, College of Applied Science, Taiyuan University of Science

and Technology. His research interest includes mathematically mechanized methods for the construction of analytical solutions to nonlinear partial differential equations and their applications.



QINQIN JIN received the B.S. degree in mathematics and applied mathematics from Shanxi Normal University, China, in 2011, the M.S. degree in computational mathematics from the University of Electronic Science and Technology of China, China, in 2014, and the Ph.D. degree in communication and information systems from Xidian University, China, in 2020. She is currently a Lecturer at the Applied Science College, Taiyuan University of Science and Technology, China. Her current

research interests include optimization algorithms, association analysis of genetic diseases, genomic signal processing, and related topics.



JIN GUO was born in Shuozhou, Shanxi, China, in 1985. She received the B.S. degree in applied chemistry from Xi'an Polytechnic University, in 2011, and the M.S. degree in materials physics and chemistry and the Ph.D. degree in materials science and engineering from the Taiyuan University of Science and Technology, in 2014 and 2019, respectively. She is currently working as an Experimentalist and the Master Director with the School of Materials Science and Engineering, Taiyuan

University of Science and Technology. Her research interests include metal compound-based new energy materials, lithium sulfur batteries, and battery performance optimization.

...