metals

Article

An Optimization Design of Energy Consumption for Aluminum Smelting Based on a Multi-Objective Artificial Vulture Algorithm

Jiayang Dai, Hangbin Liu *, Yichu Zhang, Haofan Shi and Peirun Ling

School of Electrical Engineering, Guangxi University, Nanning 530004, China; daijiayang@gxu.edu.cn (J.D.)
* Correspondence: liuhangbin@st.gxu.edu.cn

Abstract: In the process of regenerative aluminum smelting, the temperature of the furnace needs to be maintained between 700 and 850 by adjusting the setting parameters of the smelting furnace. The setting parameters are usually adjusted by manual work, and inaccuracies in manual operation can lead to wasted energy as well as unstable temperatures. Energy consumption and temperature stability are two conflicting objectives, which are difficult to find optimal parameters for the aluminum smelting process. In this paper, an improved multi-objective artificial vulture algorithm (IMOAVOA) is developed to solve a multi-objective problem of energy consumption and temperature deviations in the regenerative aluminum smelting process. The dynamic switching–elimination mechanism based on crowding distance is proposed to maintain the archive, which enhances the diversity of solutions by dynamically switching the operation space for deleting redundant solutions in the archive and dynamically deleting the solution with the smallest crowding distance in the operation space. The multi-directional leader selection mechanism is developed to select better leaders. To improve the convergence of the algorithm, the bounce strategy is introduced in the IMOAVOA. The effectiveness of the proposed algorithm is verified by UF1-UF10, kursawe, Viennet2, Viennet3, ZDT1-ZDT6, DTLZ4, and DTLZ6 test functions with several multi-objective algorithms. The experimental results indicate that IMOAVOA outperforms the original algorithm and three other multi-objective algorithms in terms of the algorithm convergence, the Pareto front coverage, and the solution diversity. Finally, the proposed algorithm is tested in an application case of regenerative aluminum smelting process. The results show that the optimal parameters for the aluminum smelting process using the proposed algorithm can reduce the consumption while meeting the objective of furnace temperature.

Keywords: aluminum smelting; optimization of energy consumption; multi-objective algorithm

check for updates

Academic Editors: Mohamad El Mehtedi and Mauro Carta

Received: 11 December 2024
Revised: 16 January 2025
Accepted: 20 January 2025
Published: 22 January 2025

Citation: Dai, J.; Liu, H.; Zhang, Y.; Shi, H.; Ling, P. An Optimization Design of Energy Consumption for Aluminum Smelting Based on a Multi-Objective Artificial Vulture Algorithm. *Metals* **2025**, *15*, 105. https://doi.org/10.3390/met15020105

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Aluminum products are widely used in various fields, such as the aerospace and automotive industries. The aluminum smelting process commonly uses regenerative smelting furnace for production. In the regenerative smelting furnace, temperature regulation and energy consumption are crucial indices that effectively save costs and reduce carbon emissions while maintaining temperature variation to meet process requirements. Reducing energy consumption and precise temperature regulation is a multi-objective problem. It is non-linear, and they interact with each other, making it difficult to find high quality solutions. It is necessary to develop a multi-objective optimization algorithm with high optimization seeking ability and good adaptability to solve such problems. In

engineering practice, multi-objective algorithms have proven to be valuable in dealing with the complexity of real-world problems. Many engineering challenges involve multiple interrelated or conflicting objectives that need to be balanced and optimized, rather than just the optimization of a single objective. For instance, improving product performance may increase costs, and reducing energy consumption may affect productivity.

In recent years, many researchers work on intelligent algorithms that can efficiently solve multi-objective problems. Multi-objective particle swarm algorithm (MOPSO) is the earlier classical multi-objective algorithm widely explored by researchers [1]. Some newly proposed algorithms have also been changed to multi-objective versions to solve multi-objective problems, such as the multi-objective ant lion optimizer (MOALO) [2], the multi-objective multi-verse optimizer (MOMVO) [3], and the multi-objective arithmetic optimization algorithm (MOAOA) [4]. A multi-objective artificial vulture optimization algorithm (MOAVOA) with external archiving and a grid mechanism is proposed by Khodadadi et al. [5]. In addition to changing the single-objective algorithms to multi-objective versions, many researchers are interested in studying the performance enhancement of multi-objective algorithms. Some researchers have worked on developing new archive maintenance methods to increase the diversity of solutions in the archive, while others have improved the convergence of the algorithms by studying new strategies or citing some methods. A multi-objective artificial hummingbird algorithm (MOAHA) is proposed by Zhao et al., where the dynamic elimination-based crowding distance (DECD) method is used to maintain an external archive to effectively preserve the population diversity [6]. A new iterative method is used for improving the convergence of the multi-objective slime mold algorithm (MOSMA) [7]. The concept of dynamic archive is proposed by Dhiman et al., characterized by caching non-dominated Pareto optimal solutions [8]. For improving convergence and inadequate constraint handling of the multi-objective particle swarm optimization algorithm (M-MOPSO) in high-dimensional problems, a dynamic boundary search strategy is proposed by Zain et al. [9], which inspired the bounce strategy in this paper. A novel multi-objective sparrow search algorithm is proposed by Dong et al., a niche optimization technology is introduced to improve the optimization effect of (MOSSA), and the levy flight strategy is introduced to enhance the ability of multi-objective sparrow search algorithm to jump out of local optimum [10]. A strengthened dominance relation method is proposed by Zouache et al. that provides a good compromise between the coverage and convergence of the obtained Pareto sets [11]. A unified space approach-based dynamic switched crowding method is proposed by Kahraman et al. to enhance the performance of multi-objective evolutionary algorithms (MOEAs) [12]. The ideas in Studies [6,12] are the inspiration for the dynamic switching–elimination mechanism based on crowding distance (DSECD) in this paper. Chaos mechanisms are also commonly used in the improvement of intelligent algorithms. An improved multi-objective manta ray foraging optimization algorithm based on Tent chaotic map and T-distribution perturbation (IMOMRFO) is proposed by Tian et al. [13]. A chaotic-based criteria is introduced to make the solutions found by the multi-objective crow search algorithm (MOCSA) more diverse [14]. Some researchers have tried to study different dominance relations to enhance the performance of multi-objective algorithms. ACOR based local search and ε-dominance strategies is used for improving the multi-objective ant colony optimization (ε-MOACOR), which produced Pareto optimal solutions with better accuracy and distribution in various benchmark tests [15].

Intelligent algorithms are very commonly used in engineering, many researchers focus on the study of intelligent algorithms and their application in engineering. Some researchers study the application of intelligent algorithms in the control optimization of industrial processes. An improved multi-objective state transition algorithm (MOSTA) is used for optimal setting control for industrial double-stream alumina digestion process [16]. Dai et al. inves-

tigate the application of multi-objective particle swarm optimization algorithm (MOPSO) for optimal control of sewage treatment process [17]. An improved multi-objective particle swarm algorithm (IMOPSO) is proposed for multi-objective optimization of stamping process parameters [18]. A novel multi-objective symbiotic organism search algorithm (MOSOS/D) for solving truss optimization problems is proposed by Kalita et al. [19]. A specialized multi-objective approach combining multi-swarm cooperative artificial bee colony (SMOABC/D) is proposed to solve the furnace-grouping problem of special aluminum ingots [20]. An improved multi-objective tuna swarm optimization (MOTSO) is proposed for the active distribution network operational optimization problem [21]. A hybrid improved moth-flame optimization with differential evolution with global and local neighborhoods algorithm (HIMD) is proposed for pose optimization on a space manipulator [22]. A sand cat algorithm incorporating learned behavior (LSCSO) is proposed by Hu et al. to unmanned aerial vehicle path planning [23]. An improved multi-objective whale algorithm is proposed by Wang et al. and apply it to the flow shop scheduling problem [24]. An improved multi-objective grasshopper algorithm (SACLMOGOA) is proposed by Wang et al., who apply it to the capacity configuration of urban rail hybrid energy storage systems [25]. An improved multi-objective aquila optimizer (IMOAO) is proposed by Nematollahi et al. to optimize the Internet of Things offloading task [26]. Lu et al. investigate the application of a hybrid multi-objective gray wolf algorithm (HMOGWO) dynamic scheduling in a real-world welding industry [27]. Fu et al. optimize the capacity configuration of CCHP system under an operating strategy by using an improved multi-objective multi-verse algorithm (IMOMVO) [28]. A hybrid strategy-based improved sparrow algorithm (HSSA) is proposed to optimize the aluminum liquid temperature prediction model in aluminum smelting process [29]. Zhang et al. investigate the application of an improved sand cat swarm optimization to the aluminum smelting process [30]. These studies provide thoughts for the research of this paper, although the researchers have made improvements to the intelligent algorithm to intensify the algorithm's ability to find the optimal, but in the face of complex multi-objective problems, there is still the possibility of falling into the local optimum. For the complex aluminum smelting process parameter optimization problem studied in this paper, it is necessary to combine the process characteristics to analyze in depth.

In summary, to solve the multi-objective optimization problem of aluminum smelting process, an improved multi-objective artificial vulture optimization algorithm (IMOAVOA) is proposed in this paper. The improved multi-objective artificial vulture algorithm is well suited for the optimal design of aluminum smelting process performance and energy consumption problems due to its powerful optimization capability and strong optimization ability for two-objective problems. The aluminum smelting process process characteristics is analyzed, and the multi-objective problem is given in the Section 2 of the article. The Section 3 of the article details the three strategies introduced in IMOAVOA, a dynamic switching–elimination mechanism based on crowding distance (DSECD) is used to maintain the IMOAVOA archive, and a multi-directional leader selection mechanism (MDLS) is developed to select a better leader for IMOAVOA. In addition, a new boundary exploration strategy called bounce strategy is introduced to enhance the exploration potential of IMOAVOA. Section 4 of the article is the experimental part, where 20 test functions are employed to evaluate the efficacy of IMOAVOA, and the IMOAVOA is applied to the multi-objective optimization problem of aluminum smelting process. Finally, Section 5 of the article is the concluding part of the paper.

2. Aluminum Smelting Process Analysis

The process of regenerative aluminum smelting is characterized as an efficient heat recovery, nonlinear process. And, aluminum smelting is a complex process that involves

several key steps. Firstly, high-temperature flames of about 1200 °C are generated by combusting fuel gas in the furnace chamber. These flames then transfer heat energy to a heat storage material, typically a ceramic ball or similar material. This helps to maintain the temperature inside the furnace by absorbing heat at high temperatures and releasing it when needed. As combustion and heating proceed, aluminum is introduced into the furnace, causing it to melt. For energy savings, it is essential to effectively control the heat storage and release of the heat storage material, which can be achieved by adjusting the combustion intensity. The aluminum smelting process may produce impurities or solid wastes that must be periodically cleaned or eliminated to ensure furnace efficiency and aluminum quality. Fine temperature regulation and process optimization are necessary for efficient aluminum smelting and to minimize energy consumption. By adjusting the fuel gas supply, combustion temperature, and heat storage material design, it is possible to optimize furnace efficiency and increase the quality of the produced aluminum.

2.1. Structure and Working Principle of Regenerative Aluminum Smelting Furnace

A regenerative aluminum smelting furnace generally has one or two pairs of regenerative burners that are symmetrically arranged in the furnace structure. The number of pairs of burners in operation depends on the smelting period: In the first and middle stages of aluminum smelting, both pairs of burners are in operation at the same time to rapidly increase the furnace temperature. In the late stage of aluminum smelting, solid aluminum is all melted into aluminum liquid, and at this time, to avoid a high-temperature aluminum liquid and reduce the burn loss, there is only one pair of burners in working condition to maintain the stability of furnace temperature. Figure 1 shows a typical regenerative aluminum smelting furnace, comprising fuel gas piping, burners, furnace chamber, exhaust piping, blower, induced draft fan, reversing valve, and other devices. For a pair of regenerative burners (A and B) in a regenerative smelt, the two burners will not be in operation at the same time. The burners load adjustment adopts the flow rate proportioning adjustment method. The burners is installed on the top of the furnace wall, and the double angle design ensures the overall coverage of the heat, making the flame and the melt have a good convection heat transfer effect. It is important to note that the two burners are not connected to the fuel gas simultaneously. When burner B is operational, the fuel gas channel in burner A is closed, and the blower circulates air through the pipe. The reversing valve is in such a state that the air/flue gas duct of burner B is fed with air: the air is blown out by the blower and flows to the reversing valve, where it is rapidly heated up to 80–90% of the furnace temperature by the heat storage area of burner B before entering the furnace chamber. Upon entering the furnace chamber, the fuel gas mixes with the air to form a gas stream with an oxygen content of less than 21%. The fuel gas then flows into the furnace chamber through the pipe and burns with the mixed gas stream, providing the necessary heat energy for smelting aluminum. The exhaust gas generated by combustion at high temperatures flows through burner A, continuously heating the ceramic ball in burner A. The ceramic ball in burner B is also heated by the ambient air, which takes away the heat. When the burner B heat storage area cannot be heated to the required temperature by the ambient air, the reversing valve will be switched to a different position, and the burner will start working. At this point, burner A's heat storage area has completed its heat storage work, and the two burners will cycle. The exhaust gas produced by smelting aluminum at high temperatures will be cooled to below 150 degrees Celsius as it passes through the heat storage area of the non-operational burner. The exhaust gas is discharged by the exhaust system of the smelt, the induced draft fan will discharge about 85% of the exhaust gas, and the remaining will be discharged by the auxiliary exhaust duct.

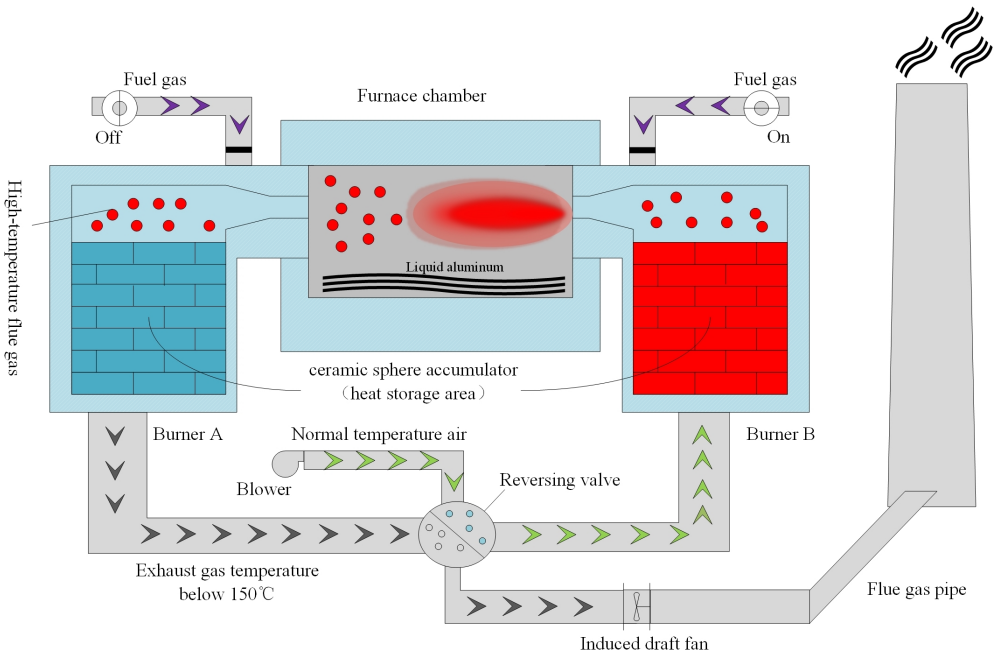


Figure 1. Working principle and structure diagram of regenerative aluminum smelting furnace.

2.2. Multi-Objective Problem Description

In aluminum smelting, in order to easily obtain the furnace temperature, a model that combines mechanistic modeling and a multi-scale kernel is introduced to obtain the furnace temperature. Furnace performance is evaluated by temperature deviation, while energy consumption serves as another index. A modified multi-objective artificial vulture optimization algorithm is used to optimize the aluminum smelting process parameters under real operating conditions. This method is developed for convenient temperature monitoring while optimizing energy consumption to increase productivity and reduce costs.

The objectives of the optimized setting parameters for the aluminum smelting process are furnace temperature and energy consumption. The multi-objective problem can be described as Equation (1),

$$\begin{cases} \text{Minimize } T_{dev}(x_i(t),t) = d_t \times abs(T_b - T_s(x_i(t),t)) \quad i = 1,2,3 \\ \text{Minimize } F_{com} = x_1(t) + x_2^3(t) \\ s.t. \quad T_s(x_i(t),t) = f(x_i(t),t) \\ \quad x_i(t) \in (X_i_\max, X_i_\min) \quad i = 1,2,3 \end{cases} \tag{1}$$

where T_{dev} represents the temperature deviation under different process setting parameters, and F_{com} represents the energy consumption under different setting parameters. T_b is the setting furnace temperature to be achieved, T_s is the temperature obtained by a model for different set parameters and the acquisition of T_s is model constrained. In addition, x_i is the set of optimization variables; specifically, x_1 to x_3 are the fuel gas flow rate, air–fuel ratio, and combustion air flow rate. X_i_max and X_i_min are the upper and lower boundaries of each variable, respectively. Furthermore, in consideration of the fact that the model does not predict temperature with absolute precision, an error correction factor, designated as d_t, is incorporated into the equation, with the value set to 10. Since the air–fuel ratio determines the energy consumed by the blower, the energy consumption is fitted by the fuel gas flow rate and the air–fuel ratio. Under the given conditions, the furnace temperature should reach T_b. Table 1 shows the recommended range of parameter settings based on the 800 °C operating conditions studied in this paper, which is determined by engineering experience.

Table 1. The recommended setting ranges for these parameters, based on experience.

Aluminum Smelting Process Parameters	Setting Range
Fuel gas flow rate	120–200 (m^3/h)
Air–fuel gas ratio	3–4
Combustion air flow rate	360–800 (m^3/h)

The equality constraint in Equation (1) needs to be obtained by some modeling methods. For convenience, a model from reference [29] is used for the equality constraint in this paper. The model is established by using a hybrid method of mechanism analysis with data-driven approach. The hybrid model can be represented by Equations (2)–(5),

$$\begin{aligned} f(x_i(t),t) = \frac{d\hat{T}(t)}{dt} = h_1 T_{rl}(t) + h_2 T_m(t) + h_3 K_{cl} + h_4 T_y(t) \cdots \\ +h_5 K_z + h_6 K_o + h_7 \hat{T}(t) + h_8 \qquad\qquad i = 1,2,3 \end{aligned} \tag{2}$$

$$K_{cl} = \sum_{j=1}^{7} \varepsilon_{clj} K_j \tag{3}$$

$$K_z = \sum_{j=1}^{7} \varepsilon_{zj} K_j \tag{4}$$

$$K_o = \sum_{j=1}^{7} \varepsilon_{oj} K_j \tag{5}$$

$$K_j = \exp\left(-\frac{\|x_i(t) - x_i(t-1)\|^2}{2\delta_j^2}\right), j = 1,2,\ldots,7, \;\; \delta_1 < \delta_2 < \cdots < \delta_7 \tag{6}$$

where T_m is the melting point of material, T_rl is the temperature of the material entering the furnace chamber, and T_y is the flue gas temperature. The three temperatures are easily accessible in the aluminum smelting process. The material discharge temperature, slag temperature and outer wall temperature are difficult to measure and are estimated by using multi-scale kernel functions. K_{cl}, K_z, and K_o are multi-scale kernels which are obtained by synthesizing seven basic kernel functions of different scales. K_j are basic kernel functions with different scales, ε_{zj}, ε_{clj}, and ε_{oj} are the weights of the basic kernel functions, and δ_j is the bandwidth of the basic kernel functions. In Equation (2), h1 to h8 are the influence coefficients to be identified. The results in reference [29] are used for the unidentified parameters of the hybrid model, as shown in Table 2.

Table 2. Model parameter settings.

Parameters	Solutions	Parameters	Solutions	Parameters	Solutions
ε_{z1}	0.6998	ε_{cl6}	0.8925	δ_4	0.6817
ε_{z2}	16.5927	ε_{cl7}	0.8597	δ_5	0.8961
ε_{z3}	0.8474	ε_{o1}	2.3681	δ_6	0.1229
ε_{z4}	2.3689	ε_{o2}	6.5071	δ_7	0.9118
ε_{z5}	0.7658	ε_{o3}	−2.9006	h_1	1.1581
ε_{z6}	0.8857	ε_{o4}	1.1679	h_2	0.0108
ε_{z7}	−26.483	ε_{o5}	1.5981	h_3	0.3687
ε_{cl1}	0.8908	ε_{o6}	1.0251	h_4	−0.1931
ε_{cl2}	0.5828	ε_{o7}	−19.067	h_5	0.7991
ε_{cl3}	1.2881	δ_1	0.8204	h_6	−37.186
ε_{cl4}	3.0199	δ_2	0.4393	h_7	1.2541
ε_{cl5}	0.7298	δ_3	0.3814	h_8	0.1684

3. Multi Objective Artificial Vulture Optimization Algorithm (MOAVOA)

The artificial vulture optimization algorithm (AVOA) is a meta-heuristic algorithm proposed by Abdollahzadeh et al. inspired by the survival behavioral patterns of African vultures [31]. The exploration strategy of the AVOA simulates the foraging and pathfinding behavior of African vultures and contains two mechanisms: exploration and exploitation. In addition, the starvation rate determines whether the vultures enter the exploration or exploitation phase, and the AVOA determines the location of the new generation of the population through the exploration strategy. The individual fitness value of each generation is evaluated by an objective function.

3.1. Multi-Objective Artificial Vulture Optimization Algorithm (MOAVOA)

Although the initial version of MOAVOA has good optimization capability, it still suffers from a lack of diversity in the solutions found and the problem of falling into local optimums. The convergence and robustness of MOAVOA also need to be further improved to adapt to the complex multi-objective problem of aluminum smelting process. The MOAVOA consists of three main major modules, which are described below:

1. An external archive mechanism is established to preserve the Pareto optimal solutions. A maximum number of solutions in the archive is set, only solutions that are not dominated by each other are allowed to exist in the archive. New solutions that want to enter the archive must not be dominated by any of the solutions in the archive, and the number of solutions in the archive is controlled to be less than or equal to the set maximum number.
2. A grid mechanism is used to maintain the archive. If the number of solutions in the archive exceeds a set value, the grid mechanism is used to re-divide the objective space in which the solutions in the archive exist into a number of sub-objective spaces, each with a corresponding index and a number of solutions distributed among them. At this point, the more solutions exist, the higher the probability that a solution in a subspace will be deleted. The redundant solutions are randomly deleted using a roulette wheel selection method in the densest region. This method balances efficiency with the diversity of solution distribution in the solution set.
3. Set up a leader selection mechanism. The leader selection mechanism also uses the grid mechanism. It selects the part of the partitioned objective space with the lowest density of solution distributions and a roulette wheel selection method is used for selecting a solution as a leader from that part. It is beneficial to strengthen the ability to explore underexplored regions and to improve the ability to jump out of local optimums of the MOAVOA.

3.2. Proposed Multi-Strategy Improved Multi-Objective Artificial Vulture Algorithm with Synergy (IMOAVOA)

In order to further improve the MOAVOA to deal with multi-objective problems, three improvements in the areas of leader selection, archive maintenance mechanism, and exploration strategy are used:

1. A multi-directional leader selection mechanism is proposed to select the best individual as the leader under the influence of multiple influencing factors by simultaneously considering the crowding distance of each solution grid space, objective space crowding distance, and decision space crowding distance in the solution set.

2. In terms of the archive maintenance mechanism, the use of the grid mechanism is abandoned, and the dynamic switching–elimination mechanism based on crowding distance (DSECD) is introduced to delete redundant non-dominated solutions in the archive, which effectively improves the distribution of solutions in the Pareto-optimal solution set.
3. A new strategy is proposed to be added into the exploration and exploitation strategy of the MOAVOA, aiming to improve the exploration capability of the MOAVOA by simulating the particle jumping behavior.

3.2.1. Multi-Directional Leader Selection Mechanism (MDLS)

In this session, the scoring selection strategy and the grid mechanism are applied to the leader of the selection IMOAVOA, where the scoring selection strategy is applied when the random number LS (between 0 and 1) is greater than 0.5, and the grid mechanism is applied when LS is less than 0.5. In the scoring selection strategy, the crowding of the objective space and the decision space of the solutions in the solution set will be calculated as an index to measure the quality of the solutions, each individual's crowding in the objective space and crowding in the decision space will be sorted to obtain the scores in both the spaces. A one-dimensional array G1 is used to store the combined scores of the individuals, the higher the combined scores represent the higher the crowding distance, which represents the higher quality of the solutions. The IMOAVOA tends to select the higher quality solution as the leader, and the probability of each solution being selected is as in Equation (7). Dynamic weights w1 are introduced when more than half of the combination scores of all the individuals are not the same. w1 is a one-dimensional array, and the number of elements in w1 is equal to the number of individuals in the archive. The size of the elements in w1 corresponds to the size of the combination scores of the individuals. At this point, the probability of each vulture individual being selected is as in Equation (8). Finally, the leader is selected randomly from among the chosen individuals and those with higher scores.

The grid mechanism is used to partition the objective space and tends to prioritize the vulture individuals in the region with the lowest solution density as the leader, and a roulette wheel selection method is used to randomly select a solution as the leader, and the probability of a vulture individual being selected in this process is as in Equation (8),

$$P_i = \frac{C}{N_i} \tag{7}$$

The number of vultures in each grid of the division is stored in a one-dimensional array called G2. To calculate the probability, dynamic weights w2 are introduced into the strategy. The size of the elements in w2 corresponds to the degree of crowding in the intervals of the divided grid. The probability of each hypercube being selected is determined by Equation (8).

$$P_i = \frac{\omega \times C}{N_i} \tag{8}$$

After the introduction of dynamic weights, the sparser the distribution of vultures the more likely that the hypercube is selected. On the contrary, the probability that the region with dense distribution of vultures is selected becomes smaller. The introduction of dynamic weights further strengthens the ability of the IMOAVOA to explore the regions with high potentials, but it is important to note that the number of individuals divided in all grids must be different by more than half to be able to introduce dynamic weights.

3.2.2. Dynamic Switching–Elimination Mechanism Based on Crowding Distance (DSECD)

Removal of redundant individuals based on the crowding distance is a mechanism used to maintain the archive, which measures the size of the crowding distance by evaluating how densely packed the individuals are in the objective space. This mechanism is used in multi-objective optimization problems to maintain the diversity of solutions in the solution set and helps the IMOAVOA to select better next generation leaders by improving the quality of solutions in the archive with the leader selection strategy, thus finding uniformly distributed solutions on the Pareto front in multi-objective optimization.

In the IMOAVOA, the use of the grid mechanism for archive maintenance is abandoned in favor of a DSECD mechanism for archive maintenance. The DSECD mechanism considers not only the crowding distance of the individuals in the objective space, but also the crowding distance of the individuals in the decision space. It means that there are two operation spaces for deleting individuals using dynamic crowding distance and dynamic selection. In addition, a signal-triggering mechanism is employed to select the operating space between the decision space and the objective space to maintain the archive. After the selection is complete, IMOAVOA employs dynamic elimination to remove individuals with low crowding distance in the operational space. As the first solution with the lowest crowding distance is removed, the crowding distance of the two neighboring solutions in the archive change, the crowding distances of these two solutions need to be recalculated to update the archive. And then the individual with the lowest crowding in the already updated archive was again identified and removed, and this step was repeated until all redundant solutions in the archive were removed. Tent sequence chaotic sequence is used as the trigger mechanism for this modification, and the selection process is shown in in Equation (9),

$$SP = \begin{cases} D_{sp} & \text{if } iter < 0.2 max_iter \&\& T(r) < 0.5 \\ O_{sp} & \text{if } iter >= 0.2 max_iter \,||\, T(r) >= 0.5 \end{cases} \tag{9}$$

where SP is the selected space, D_{sp} is the decision space, and O_{sp} is the objective space. $iter$ is the current iteration number, max_iter is the maximum iteration number, T is a tent chaotic sequence of one hundred sequence numbers, and r is a random number between 1 and 100.

3.2.3. Bounce Strategy with Positional Offset Probability

The MOAVOA is likely to enter a stagnation period of exploration when the number of iterations is relatively high and the MOAVOA cannot find a new and better solution. In order to improve the exploration performance of the MOAVOA, a bounce strategy with positional offset probability is added to the original exploration process. The bounce strategy can be started when the IMOAVOA iteration exceeds half of the set maximum number of iterations. In the bounce strategy, the change of solutions in the archive between the previous generation and the current generation is first judged; if the number of solutions in the two solution sets is the same and the number of difference solutions between the new solution set and the old solution set is less than 0.05 times the number of the current archive, then the bounce strategy is applied to generate an upward or downward perturbation. The positional offset probability P_{mi} is computed by Equation (10), and the bounce strategy is executed by Equation (12).

$$P_{mi} = \frac{d_{1i}}{d_{1i} + d_{2i}} \tag{10}$$

$$\begin{cases} d_{1i} = \sqrt{\sum_{j=1}^{N} (ub(i,j) - V(i,j))^2} \\ d_{2i} = \sqrt{\sum_{j=1}^{N} (lb(i,j) - V(i,j))^2} \end{cases} \tag{11}$$

$$V'(i,j) = \begin{cases} \text{Eq(13)if} P_{mi} < 0.5 \\ \text{Eq(14)if} P_{mi} >= 0.5 \end{cases} \tag{12}$$

$$V'(i,j) = B(i,j) + 4\ln(e - 1.7\text{rand}_1) \times (\frac{\text{iter}}{\text{max_iter}}) \times \text{SR} \tag{13}$$

$$V'(i,j) = B(i,j) - 4\ln(e - 1.7\text{rand}_2) \times (\frac{\text{iter}}{\text{max_iter}}) \times \text{SR} \tag{14}$$

$$\text{SR} = |T(r) \times B(i,j) - V(i,j)| \times |0.5 - \text{rand}| \tag{15}$$

In Equation (10)–(15), d_{1i} and d_{2i} are the Euclidean distances between the current individual position and the upper and lower boundaries, which are calculated from Equation (11), ub is the upper boundary, lb is the lower boundary, and $V(i)$ is the current position of the vulture. $V'(i)$ is the updated position, B is BestVulture2, and rand is a random number between 0 and 1.

3.3. Computational Complexity

In the IMOAVOA, the number of individuals (N) in the population affects the computational complexity. Larger populations may require more computational resources and time to execute. In addition, the maximum number of iterations (M), the number of objective functions (Q), the number of archives, and the dimension of variables (D) are also the main factors affecting the computational complexity of the IMOAVOA, and the number of archives is set to be consistent with the number of populations.

The computational complexity of the leader selection mechanism is O($0.5QMN^2 + 0.5QMN$).

The process of archive maintenance mechanism using DSECD mechanism is divided into the non-dominated sorting of individuals in the population after archiving, calculating the crowding distance between two and two individuals in the objective space or decision space and sorting to delete the redundant individuals. The computational complexity of this process is O($QMN^2logN + 2QMN$).

The computational complexity of the Exploration and Exploitation phases is O($0.5QMN$) and O($0.5QMN$), respectively.

The computational complexity of the bounce strategy is O($0.5QMN^3D$).

Thus, the computational complexity of the IMOAVOA in considering the worst case is max{O($0.5QMN^2 + 0.5QMN$), O($QMN^2logN + 2QMN$), O(QMN), O($0.5QMN^3D$)}, i.e., O(IMOAVOA) = O(QMN^3D).

4. Experimental Design and Analysis of Results

In the experiment, each function will be repeated 40 times. The mean and standard deviation of each metric are presented in the table for comparison, with underlining indicating the best performance. Original MOAVOA added to MDLS to become SMOAVOA, JMOAVOA is based on SMOAVOA with a bounce strategy, and IMOAVOA is the version that introduces all the improved strategies.

4.1. Experimental Setup

Unconstrained and constrained bi-objective and tri-objective math problems are used as a test set for verifying the ability of the IMOAVOA to handle non-convex and non-linear problems. The test set UF1-UF10 [32], Kursawe [33], Viennet2, Viennet3 [34], ZDT1-ZDT6 [35], DTLZ4, and DTLZ6 [36] test functions are used to validate the performance of the IMOAVOA. Test Functions UF1 through UF10 represent a series of ten unconstrained, multi-objective optimization problems characterized by complex, non-convex Pareto fronts and multiple conflicting objectives. These problems are designed to evaluate the performance of the algorithms in dealing with multimodal versus nonconvex problems. The Kursawe test function is a bi-objective optimization problem that is primarily used to examine the algorithm's ability to handle complex interactions between decision variables. Viennet2 and Viennet3, while simpler in structure, are still challenging because they test the algorithm's ability to balance convergence and diversity in the face of conflicting objectives. The ZDT1 through ZDT6 families grow in complexity, with each test function designed to evaluate specific features of multi-objective optimization such as convexity, non-convexity, and the ability to operate in higher dimensional spaces. DTLZ4 and DTLZ6 are both tri-objective test functions designed to test the algorithm's ability to scale a large number of objectives and decision variables. The computer configuration and software used during the experiments are as follows: 64-bit Windows 11 Professional operating system, Microsoft Corporation, Redmond, WA, USA. X64-based processor with 8.00 GB of RAM, Asus, Taipei, Taiwan, China. Processor 12th Gen Intel(R) Core (TM) i5-12400 2.50 GHz, Intel Corporation, Santa Clara, CA, USA. Matlab R2019b (9.7. 0.1190202), MathWorks, Natick, MA, USA.

4.2. Algorithms and Parameter Settings for Comparison

In order to verify the performance of IMOAVOA, MOPSO [1], MOALO [2], and MOAOA [4] are used to make a comparison, and the distribution of the Pareto optimal frontiers of several algorithms will be given in this section. For objectivity, the parameters used for comparison in the algorithms are set according to the original reference of each algorithm. The detailed parameter settings for each algorithm are shown in Table 3.

Table 3. Parameters for different multi-objective optimization algorithms.

Parameters	MOPSO	MOALO	MOAOA	MOAVOA
Mutation Probability (Pw; or pro)	0.5	–	–	–
Population Size (Npop)	100	100	100	100
Archive Size (Nrep; or TM)	100	100	100	100
Number of Adaptive Grid (Ngrid)	30	30	30	30
Personal Learning Coefficient (C1)	1	–	–	–
Global Learning Coefficient (C2)	2	–	–	–
Inertia Weight (w)	0.4	–	–	–
Beta	4	4	4	4
Gamma	2	2	2	2

4.3. Performance Metrics

In this section, inverted generational distance (IGD), generational distance (GD), and Hypervolume (HV) total three metrics are used to evaluate the results of IMOAVOA.

4.3.1. Inverted Generational Distance (IGD)

IGD is calculated by averaging the minimum distance from each solution to the known Pareto front in the set of solutions generated by the algorithm, i.e., the smaller the distance, the greater the contribution of the solution. A lower value of IGD indicates that the algorithm is closer to the known Pareto front, meaning that the algorithm has a superior performance is calculated by Equation (16).

$$\mathrm{IGD} = \frac{\sqrt{\sum_{i=1}^{n} d_i^2}}{n} \tag{16}$$

4.3.2. Generational Distance (GD)

GD is used to measure the distance between the set of solutions generated by the algorithm and the true Pareto frontiers to evaluate its convergence and the quality of the generated solutions. GD is calculated by Equation (17).

$$\mathrm{GD} = \left(\frac{1}{n_{\mathrm{pf}}} \sum_{i=1}^{n_{\mathrm{pf}}} \mathrm{dis}_i^2 \right)^{\frac{1}{2}} \tag{17}$$

4.3.3. Hypervolume (HV)

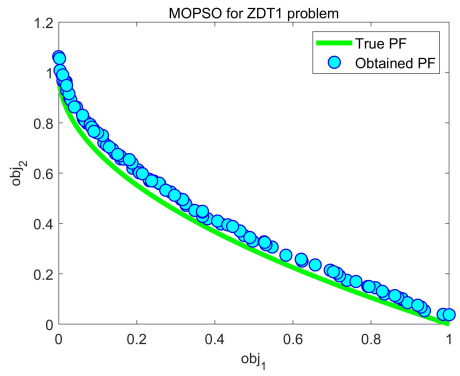
HV is mainly used to evaluate the coverage and diversity of the solution set generated by the algorithm in the objective space; the larger the value of HV, the better the performance of the algorithm. Let the s vector denote the minimum point in the objective space. Define a z-vector in the w-space to denote the PFE estimated by the algorithm. HV is calculated by Equation (18).

$$HV(s, W) = \lambda_n(AptCommand22C3;_{z \in W}[s; W]) \tag{18}$$

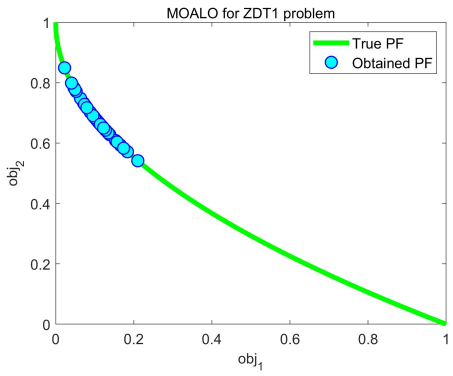
where W $\subset$ R_n, for all z $\in$ W and z$\prec$s in this space and λ_n is the n-dimensional Lebesgue measure.

4.4. Results and Analysis

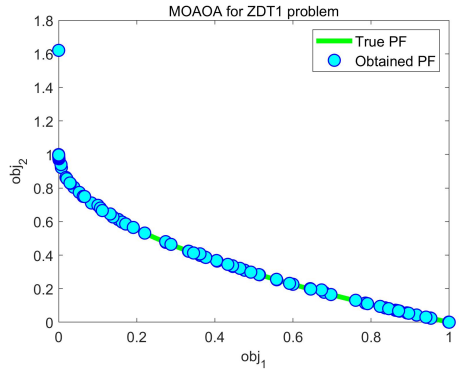
Figures 2 and 3 show the performance of MOPSO, MOALO, MOAOA and IMOAVOA on the ZDT1 and ZDT3 test functions, and IMOAVOA outperforms the other three algorithms in terms of solution distribution and convergence. Table 4 shows the performance of MOPSO, MOALO, MOAOA, and IMOAVOA on the IGD metrics. This IGD metric comprehensively evaluates the performance of the four algorithms, taking into account both large distributional spreads and large Euclidean distances from the nearest true value. A smaller calculated value indicates better convergence and distributional spreads. IMOAVOA performs extremely well on the IGD metrics, outperforming the other three algorithms on the mean of the IGD metrics over the 19 test functions, and only performs less well than MOPSO on the UF8 test function. The experimental data indicate that MOPSO performs slightly worse than IMOAVOA on functions UF2 and UF4 in terms of mean, but its standard deviation is lower than that of IMOAVOA, which suggests that the smoothness is better on UF2 and UF4, although the difference is minimal. MOAOA performs worse than IMOAVOA on UF10 in terms of mean, but its stability is slightly better than that of the latter.



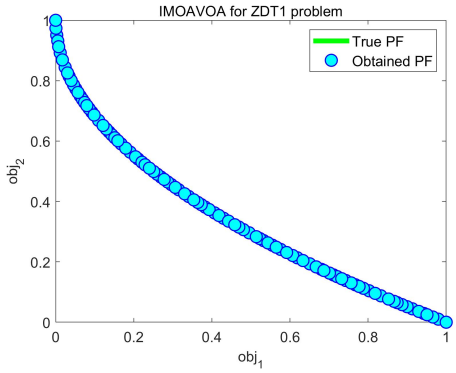
(**a**) MOPSO for ZDT1 problem



(**b**) MOALO for ZDT1 problem

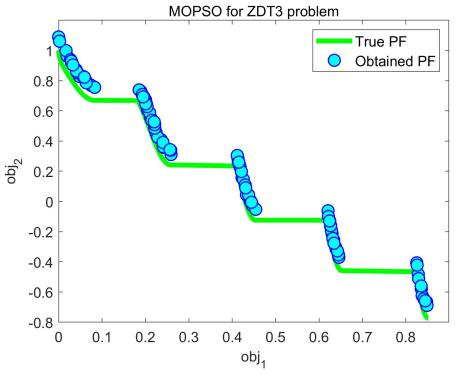


(**c**) MOAOA for ZDT1 problem

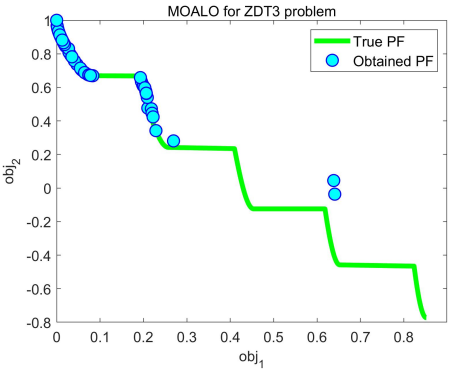


(**d**) IMOAVOA for ZDT1 problem.

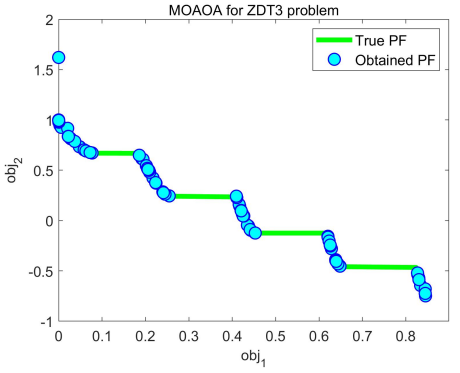
Figure 2. Comparison of different algorithms for ZDT1 problem.



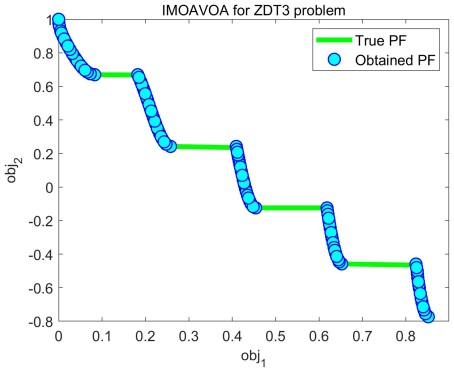
(**a**) MOPSO for ZDT3 problem



(**b**) MOALO for ZDT3 problem



(**c**) MOAOA for ZDT3 problem



(**d**) IMOAVOA for ZDT3 problem

Figure 3. Comparison of different algorithms for ZDT3 problem.

Table 5 displays the performance on the GD metrics of multi-objective algorithms. The goodness of the GD metric can be an intuitive reflection of the ability of the multi-objective algorithms to explore the search for excellence. IMOAVOA exhibits relatively good performance in the 12 test functions, particularly in the GD metrics, where it performs exceptionally well. Among these functions, IMOAVOA outperforms the other three comparison algorithms in the GD metrics. However, IMOAVOA does not perform as well as the other comparison algorithms in the other eight test functions. Among these functions, IMOAVOA shows relatively poor results in the GD metrics, but only a slight disadvantage in five of the eight poorly performing functions. IMOAVOA performs well in the ZDT and UF series, particularly in ZDT3, ZDT4, UF1, UF5, UF6, UF7, Kursawe test functions, and the Viennet3 test function, where it demonstrates strong and stable optimality search. IMOAVOA outperforms MOPSO and MOALO on ZDT6 test function, but not MOAOA. However, MOPSO performs exceptionally well on DTLZ4 and UF4 test functions, with a large lead in both the mean and the standard deviation. Additionally, MOAOA performs relatively well on the mean on DTLZ6 test function, but not as well as IMOAVOA in the standard deviation. IMOAVOA performs better on functions UF2 and UF3 test functions, but has higher standard deviations compared to MOPSO and MOALO, respectively. This suggests that the results obtained by IMOAVOA on UF2 and UF3 test functions may not be very stable, and that there is a significant difference between the best and worst results among the 40 trials. MOALO performs exceptionally well on functions UF8, UF9, and Viennet3 test functions, but has a higher standard deviation on the UF8 test function compared to MOAOA.

Table 6 illustrates the performance of the four algorithms based on the HV metrics. IMOAVOA demonstrates outstanding performance in 14 test functions. Within the ZDT test function series, it only falls behind MOALO in ZDT3 test function and in the 14 metrics with excellent HV scores. The findings indicate IMOAVOA's strong performance across both GD and IGD metrics, suggesting a uniform distribution of solutions in space and excellent distribution on the Pareto front. The IMOAVOA consistently identifies high-quality solutions closer to the true Pareto front. While Pareto-optimal solutions obtained by MOALO for ZDT3 test function occupy significant space, its performance in GD and IGD metrics is subpar, suggesting that although solutions are widely distributed, they may not be optimal. However, MOALO outperforms other three algorithms on the Kursawe test function, with superior HV and GD metrics. Moreover, the solution set produced by MOALO on Kursawe test function closely approximates the ideal Pareto solution set distribution. MOPSO exhibits widely distributed Pareto-optimal solution sets for UF5, UF8, and Viennet2 test functions but weaker GD and IGD metrics compared to IMOAVOA on these test functions. Thus, MOPSO excels in finding global optima, with superior uniformity in solution distribution on these functions. MOAOA performs strong performance in HV metric testing for UF10 test function.

Tables 7–9 shows the changes in IGD, GD, and HV metrics of the multi-objective artificial vulture optimization algorithm after sequentially overlaying the MDLS mechanism, the bounce strategy, and the DSECD mechanism. The performance of the IGD metric of MOAVOA is significantly improved after the addition of the DSECD mechanism, which indicates that the addition of the DSECD mechanism greatly improves the ability of MOAVOA to find solution diversity and enhances its smoothness and robustness. The data in Table 7 show that the average values of the IGD metrics for the 12 test functions of MOAVOA are improved after the addition of the MDLS mechanism. JMOAVOA achieves an advantage in the mean value of IGD metrics for 14 test functions over SMOAVOA after the addition of the bounce strategy. The data in Table 8 show that the addition of the DSECD mechanism also improves the convergence of MOAVOA. The MDLS mechanism improves the performance of ZDT1-ZDT3, DTLZ4, DTLZ6, UF1-UF8, UF10, and Kursawe test functions compared to MOAVOA. And, after adding the bounce strategy to the multi-leader selection strategy, the JMOAVOA performs

amazingly well on the ZDT series of test functions with a great degree of improvement, showing strong convergence on the ZDT1-ZDT6 test functions. In addition, with the inclusion of the bounce strategy, JMOAVOA performs better than MOAVOA and SMOAVOA on the UF1, UF3-UF7, UF9, and Viennet3 test functions. Table 9 shows that IMOAVOA maintains the lead in the mean of HV metrics on 16 test functions, so the introduction of the DSECD mechanism has a significant improvement on the performance of JMOAVOA on HV metrics. The analysis of the standard deviation of SMOAVOA shows that it outperforms MOAVOA on 17 test functions, indicating that the MDLS mechanism is very stable in improving the MOAVOA's ability to find global solutions. By introducing the bounce strategy, the HV metrics of the JMOAVOA are improved, especially on the ZDT3 and UF5 test functions, where the mean values are better than MOAVOA, SMOAVOA and IMOAVOA.

Table 4. Results of IGD metric on benchmark functions.

Fun	MOPSO		MOALO		MOAOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	1.24×10^{-3}	1.16×10^{-4}	1.75×10^{-2}	3.13×10^{-3}	3.96×10^{-4}	4.87×10^{-5}	1.46×10^{-4}	1.17×10^{-6}
ZDT2	1.13×10^{-3}	2.37×10^{-4}	2.03×10^{-2}	2.55×10^{-3}	5.10×10^{-4}	1.18×10^{-4}	1.50×10^{-4}	1.89×10^{-6}
ZDT3	9.52×10^{-4}	1.27×10^{-4}	5.75×10^{-3}	2.13×10^{-3}	3.26×10^{-4}	4.79×10^{-5}	1.06×10^{-4}	1.94×10^{-6}
ZDT4	6.48×10^{-2}	2.26×10^{-2}	2.91×10^{-2}	3.38×10^{-2}	6.10×10^{-4}	1.18×10^{-4}	1.47×10^{-4}	1.30×10^{-6}
ZDT6	4.73×10^{-2}	3.35×10^{-2}	5.87×10^{-3}	6.10×10^{-3}	6.10×10^{-4}	1.35×10^{-4}	1.44×10^{-4}	3.16×10^{-6}
DTLZ4	3.92×10^{-3}	9.28×10^{-4}	1.23×10^{-2}	5.69×10^{-3}	9.12×10^{-3}	3.65×10^{-3}	2.40×10^{-3}	1.08×10^{-4}
DTLZ6	4.19×10^{-2}	9.59×10^{-3}	8.40×10^{-3}	3.76×10^{-3}	1.33×10^{-3}	4.06×10^{-4}	2.03×10^{-4}	4.26×10^{-6}
UF1	8.21×10^{-4}	2.58×10^{-4}	2.46×10^{-3}	5.12×10^{-4}	1.58×10^{-3}	2.66×10^{-4}	5.71×10^{-4}	1.10×10^{-4}
UF2	4.63×10^{-4}	3.97×10^{-5}	1.46×10^{-3}	4.35×10^{-4}	9.84×10^{-4}	1.60×10^{-4}	3.36×10^{-4}	4.09×10^{-5}
UF3	6.25×10^{-3}	1.48×10^{-3}	1.08×10^{-2}	1.43×10^{-3}	8.52×10^{-3}	2.96×10^{-3}	5.85×10^{-3}	1.18×10^{-3}
UF4	1.22×10^{-3}	4.46×10^{-5}	2.59×10^{-3}	2.32×10^{-4}	1.19×10^{-3}	4.75×10^{-5}	1.15×10^{-3}	4.58×10^{-5}
UF5	5.77×10^{-2}	3.47×10^{-2}	1.36×10^{-1}	4.38×10^{-2}	1.10×10^{-1}	3.11×10^{-2}	5.43×10^{-2}	1.09×10^{-2}
UF6	1.86×10^{-2}	5.49×10^{-3}	2.79×10^{-2}	9.90×10^{-3}	3.58×10^{-2}	7.50×10^{-3}	1.36×10^{-2}	3.66×10^{-3}
UF7	1.04×10^{-3}	3.94×10^{-4}	6.02×10^{-3}	3.62×10^{-3}	1.27×10^{-3}	3.60×10^{-4}	5.33×10^{-4}	1.25×10^{-4}
UF8	4.52×10^{-3}	4.77×10^{-4}	1.99×10^{-2}	1.09×10^{-2}	7.74×10^{-3}	2.16×10^{-3}	4.64×10^{-3}	8.37×10^{-4}
UF9	6.49×10^{-3}	2.09×10^{-3}	2.54×10^{-2}	1.27×10^{-2}	1.18×10^{-2}	3.96×10^{-3}	5.12×10^{-3}	1.06×10^{-3}
UF10	1.66×10^{-2}	2.98×10^{-3}	3.16×10^{-2}	7.25×10^{-3}	1.47×10^{-2}	2.39×10^{-3}	1.24×10^{-2}	3.32×10^{-3}
Kursawe	2.56×10^{-4}	3.31×10^{-5}	1.32×10^{-3}	5.89×10^{-4}	1.47×10^{-3}	9.15×10^{-4}	1.41×10^{-4}	3.82×10^{-6}
Viennet2	3.64×10^{-4}	2.94×10^{-4}	2.85×10^{-3}	3.90×10^{-4}	6.19×10^{-4}	2.14×10^{-4}	3.08×10^{-4}	3.25×10^{-5}
Viennet3	2.11×10^{-4}	5.41×10^{-5}	1.85×10^{-3}	1.33×10^{-3}	9.22×10^{-4}	9.44×10^{-4}	1.84×10^{-4}	3.56×10^{-5}

Note: The underline indicates the best performance.

Table 5. Results of GD metric on benchmark functions.

Fun	MOPSO		MOALO		MOAOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	4.08×10^{-3}	3.88×10^{-4}	1.87×10^{-4}	3.70×10^{-4}	3.34×10^{-3}	2.38×10^{-3}	1.27×10^{-4}	2.46×10^{-5}
ZDT2	3.66×10^{-3}	7.98×10^{-4}	2.99×10^{-5}	4.44×10^{-6}	2.35×10^{-3}	2.82×10^{-3}	4.62×10^{-5}	2.18×10^{-6}
ZDT3	2.99×10^{-3}	4.74×10^{-4}	2.06×10^{-3}	1.87×10^{-3}	7.40×10^{-3}	4.50×10^{-3}	1.00×10^{-4}	5.93×10^{-6}
ZDT4	2.28×10^{-1}	8.14×10^{-2}	2.55×10^{-2}	1.61×10^{-1}	1.73×10^{-4}	3.03×10^{-5}	6.98×10^{-5}	7.70×10^{-6}
ZDT6	2.59×10^{-1}	2.48×10^{-1}	8.76×10^{-2}	4.94×10^{-2}	1.01×10^{-2}	9.47×10^{-3}	5.65×10^{-2}	4.64×10^{-2}
DTLZ4	2.20×10^{-3}	6.63×10^{-4}	6.20×10^{-3}	3.94×10^{-3}	4.75×10^{-3}	2.12×10^{-3}	3.29×10^{-3}	1.08×10^{-3}
DTLZ6	2.74×10^{-1}	2.90×10^{-2}	1.16×10^{-1}	9.02×10^{-2}	3.02×10^{-5}	2.83×10^{-6}	5.97×10^{-5}	2.55×10^{-6}
UF1	1.14×10^{-2}	1.53×10^{-2}	2.18×10^{-2}	3.35×10^{-2}	6.33×10^{-2}	4.08×10^{-2}	7.42×10^{-3}	7.85×10^{-3}
UF2	6.90×10^{-3}	7.34×10^{-3}	3.81×10^{-2}	2.96×10^{-2}	1.45×10^{-2}	1.27×10^{-2}	5.39×10^{-3}	9.14×10^{-3}
UF3	5.49×10^{-2}	9.14×10^{-2}	1.01×10^{-2}	2.03×10^{-3}	2.43×10^{-2}	5.31×10^{-2}	7.96×10^{-3}	3.36×10^{-3}
UF4	4.56×10^{-3}	4.86×10^{-4}	1.12×10^{-2}	2.80×10^{-3}	4.88×10^{-3}	5.78×10^{-4}	4.57×10^{-3}	6.39×10^{-4}
UF5	1.18×10^{-1}	1.39×10^{-1}	2.47×10^{-1}	1.52×10^{-1}	7.16×10^{-1}	2.93×10^{-1}	6.46×10^{-2}	4.56×10^{-2}
UF6	6.04×10^{-1}	5.46×10^{-1}	4.48×10^{-1}	2.64×10^{-1}	1.47×10^{0}	6.93×10^{-1}	1.99×10^{-1}	1.27×10^{-1}
UF7	8.31×10^{-3}	1.25×10^{-2}	2.52×10^{-2}	2.31×10^{-2}	5.16×10^{-2}	3.51×10^{-2}	6.69×10^{-3}	9.77×10^{-3}
UF8	3.53×10^{-1}	1.85×10^{-1}	1.46×10^{-1}	1.79×10^{-1}	3.25×10^{-1}	7.44×10^{-2}	3.08×10^{-1}	1.80×10^{-1}
UF9	3.69×10^{-1}	1.92×10^{-1}	9.23×10^{-2}	9.50×10^{-2}	4.90×10^{-1}	1.16×10^{-1}	2.09×10^{-1}	1.03×10^{-1}
UF10	1.44×10^{0}	6.78×10^{-1}	1.09×10^{0}	3.50×10^{-1}	1.87×10^{0}	5.73×10^{-1}	1.05×10^{0}	6.16×10^{-1}
Kursawe	4.38×10^{-3}	9.20×10^{-4}	2.12×10^{-2}	1.61×10^{-2}	1.64×10^{-2}	2.02×10^{-2}	1.91×10^{-4}	2.25×10^{-5}
Viennet2	9.46×10^{-4}	4.15×10^{-4}	1.45×10^{-4}	2.05×10^{-4}	5.46×10^{-3}	7.25×10^{-3}	8.50×10^{-4}	3.45×10^{-4}
Viennet3	7.45×10^{-4}	3.21×10^{-4}	3.35×10^{-3}	7.57×10^{-3}	1.41×10^{-3}	3.57×10^{-4}	2.31×10^{-4}	8.05×10^{-5}

Note: The underline indicates the best performance.

Table 6. Results of HV metric on benchmark functions.

Fun	MOPSO		MOALO		MOAOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	6.70×10^{-1}	4.75×10^{-3}	4.35×10^{-1}	6.81×10^{-2}	7.14×10^{-1}	1.07×10^{-3}	7.20×10^{-1}	2.15×10^{-5}
ZDT2	3.97×10^{-1}	1.01×10^{-2}	1.05×10^{-1}	2.18×10^{-2}	4.38×10^{-1}	1.01×10^{-3}	4.45×10^{-1}	2.22×10^{-5}
ZDT3	5.76×10^{-1}	4.74×10^{-3}	7.20×10^{-1}	3.83×10^{-2}	5.99×10^{-1}	4.79×10^{-3}	6.00×10^{-1}	2.45×10^{-5}
ZDT4	4.23×10^{-3}	1.86×10^{-2}	2.58×10^{-1}	8.27×10^{-2}	7.09×10^{-1}	2.88×10^{-3}	7.20×10^{-1}	2.36×10^{-5}
ZDT6	9.04×10^{-3}	2.02×10^{-2}	3.10×10^{-1}	6.92×10^{-2}	3.80×10^{-1}	2.29×10^{-3}	3.89×10^{-1}	8.06×10^{-5}
DTLZ4	5.09×10^{-1}	1.76×10^{-2}	3.48×10^{-1}	1.15×10^{-1}	4.72×10^{-1}	5.54×10^{-2}	5.35×10^{-1}	3.99×10^{-3}
DTLZ6	2.27×10^{-3}	1.44×10^{-2}	1.18×10^{-1}	2.63×10^{-2}	1.91×10^{-1}	1.48×10^{-3}	2.00×10^{-1}	3.53×10^{-5}
UF1	6.94×10^{-1}	7.86×10^{-3}	6.22×10^{-1}	2.42×10^{-2}	6.64×10^{-1}	8.72×10^{-3}	7.04×10^{-1}	3.42×10^{-3}
UF2	7.07×10^{-1}	1.36×10^{-3}	6.72×10^{-1}	9.82×10^{-3}	6.90×10^{-1}	4.45×10^{-3}	7.13×10^{-1}	1.02×10^{-3}
UF3	4.74×10^{-1}	5.50×10^{-2}	2.90×10^{-1}	4.45×10^{-2}	3.86×10^{-1}	7.99×10^{-2}	5.04×10^{-1}	4.14×10^{-2}
UF4	3.94×10^{-1}	1.90×10^{-3}	3.30×10^{-1}	1.07×10^{-2}	3.96×10^{-1}	1.86×10^{-3}	3.99×10^{-1}	1.88×10^{-3}
UF5	2.91×10^{-1}	8.04×10^{-2}	6.59×10^{-2}	7.18×10^{-2}	5.96×10^{-2}	4.94×10^{-2}	2.57×10^{-1}	5.31×10^{-2}
UF6	1.51×10^{-1}	7.61×10^{-2}	5.01×10^{-2}	6.85×10^{-2}	6.20×10^{-3}	2.12×10^{-2}	1.64×10^{-1}	8.30×10^{-2}
UF7	5.50×10^{-1}	1.22×10^{-2}	4.11×10^{-1}	6.72×10^{-2}	5.35×10^{-1}	1.37×10^{-2}	5.69×10^{-1}	3.22×10^{-3}
UF8	4.38×10^{-1}	2.07×10^{-2}	1.47×10^{-1}	9.30×10^{-2}	3.99×10^{-1}	3.92×10^{-2}	4.34×10^{-1}	3.21×10^{-2}
UF9	6.55×10^{-1}	4.46×10^{-2}	2.81×10^{-1}	1.73×10^{-1}	5.10×10^{-1}	9.55×10^{-2}	6.90×10^{-1}	3.35×10^{-2}
UF10	1.06×10^{-1}	7.34×10^{-2}	1.55×10^{-2}	4.02×10^{-2}	3.02×10^{-1}	4.03×10^{-2}	2.47×10^{-1}	8.57×10^{-2}
Kursawe	5.11×10^{-1}	1.86×10^{-2}	5.47×10^{-1}	4.13×10^{-2}	4.98×10^{-1}	1.53×10^{-2}	5.04×10^{-1}	3.36×10^{-4}
Viennet2	3.39×10^{-1}	1.61×10^{-3}	3.23×10^{-1}	9.08×10^{-3}	3.37×10^{-1}	1.08×10^{-3}	3.38×10^{-1}	2.17×10^{-4}
Viennet3	1.83×10^{-1}	2.54×10^{-4}	1.58×10^{-1}	5.08×10^{-2}	1.82×10^{-1}	2.80×10^{-4}	1.83×10^{-1}	8.34×10^{-5}

Note: The underline indicates the best performance.

Table 7. Results of IGD metric on benchmark functions.

Fun	MOAVOA		SMOAVOA		JMOAVOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	4.26×10^{-4}	1.03×10^{-4}	4.25×10^{-4}	4.61×10^{-5}	5.04×10^{-4}	3.40×10^{-4}	1.46×10^{-4}	1.17×10^{-6}
ZDT2	4.38×10^{-4}	7.14×10^{-5}	4.36×10^{-4}	6.91×10^{-5}	4.37×10^{-4}	1.35×10^{-4}	1.50×10^{-4}	1.89×10^{-6}
ZDT3	3.16×10^{-4}	5.36×10^{-5}	3.01×10^{-4}	3.40×10^{-5}	3.87×10^{-4}	3.85×10^{-4}	1.06×10^{-4}	1.94×10^{-6}
ZDT4	4.46×10^{-4}	6.81×10^{-5}	4.33×10^{-4}	6.79×10^{-5}	4.66×10^{-4}	1.07×10^{-4}	1.47×10^{-4}	1.30×10^{-6}
ZDT6	3.60×10^{-4}	3.74×10^{-5}	3.66×10^{-4}	3.91×10^{-5}	3.73×10^{-4}	3.32×10^{-5}	1.44×10^{-4}	3.16×10^{-6}
DTLZ4	4.08×10^{-3}	8.00×10^{-4}	4.20×10^{-3}	8.97×10^{-4}	4.54×10^{-3}	1.19×10^{-3}	2.40×10^{-3}	1.08×10^{-4}
DTLZ6	6.04×10^{-4}	1.19×10^{-4}	5.64×10^{-4}	9.20×10^{-5}	5.93×10^{-4}	1.54×10^{-4}	2.03×10^{-4}	4.26×10^{-6}
UF1	9.40×10^{-4}	1.74×10^{-4}	9.61×10^{-4}	1.34×10^{-4}	7.97×10^{-4}	1.10×10^{-4}	5.71×10^{-4}	1.10×10^{-4}
UF2	8.92×10^{-4}	2.61×10^{-4}	8.68×10^{-4}	1.97×10^{-4}	7.43×10^{-4}	1.15×10^{-4}	3.36×10^{-4}	4.09×10^{-5}
UF3	6.69×10^{-3}	1.42×10^{-3}	6.82×10^{-3}	1.13×10^{-3}	6.27×10^{-3}	1.42×10^{-3}	5.85×10^{-3}	1.18×10^{-3}
UF4	1.48×10^{-3}	1.27×10^{-4}	1.47×10^{-3}	8.38×10^{-5}	1.49×10^{-3}	7.93×10^{-5}	1.15×10^{-3}	4.58×10^{-5}
UF5	6.56×10^{-2}	1.36×10^{-2}	6.51×10^{-2}	1.21×10^{-2}	5.44×10^{-2}	1.38×10^{-2}	5.43×10^{-2}	1.09×10^{-2}
UF6	1.68×10^{-2}	3.96×10^{-3}	1.76×10^{-2}	4.94×10^{-3}	1.49×10^{-2}	3.88×10^{-3}	1.36×10^{-2}	3.66×10^{-3}
UF7	8.16×10^{-4}	1.66×10^{-4}	8.04×10^{-4}	1.24×10^{-4}	7.96×10^{-4}	1.12×10^{-4}	5.33×10^{-4}	1.25×10^{-4}
UF8	5.26×10^{-3}	8.92×10^{-4}	5.97×10^{-3}	1.48×10^{-3}	5.45×10^{-3}	1.31×10^{-3}	4.64×10^{-3}	8.37×10^{-4}
UF9	5.63×10^{-3}	9.25×10^{-4}	5.55×10^{-3}	8.02×10^{-4}	5.69×10^{-3}	1.12×10^{-3}	5.12×10^{-3}	1.06×10^{-3}
UF10	1.53×10^{-2}	3.56×10^{-3}	1.36×10^{-2}	2.64×10^{-3}	1.40×10^{-2}	3.20×10^{-3}	1.24×10^{-2}	3.32×10^{-3}
Kursawe	4.47×10^{-4}	9.57×10^{-5}	4.74×10^{-4}	2.06×10^{-4}	4.24×10^{-4}	5.62×10^{-5}	1.41×10^{-4}	3.82×10^{-6}
Viennet2	2.98×10^{-4}	3.98×10^{-5}	3.21×10^{-4}	5.42×10^{-5}	3.00×10^{-4}	3.93×10^{-5}	3.08×10^{-4}	3.25×10^{-5}
Viennet3	4.95×10^{-4}	2.83×10^{-4}	4.60×10^{-4}	2.00×10^{-4}	4.29×10^{-4}	1.79×10^{-4}	1.84×10^{-4}	3.56×10^{-5}

Note: The underline indicates the best performance.

Table 8. Results of GD metric on benchmark functions.

Fun	MOAVOA		SMOAVOA		JMOAVOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	7.15×10^{-4}	3.49×10^{-3}	1.69×10^{-4}	2.14×10^{-2}	1.31×10^{-4}	7.73×10^{-5}	1.27×10^{-4}	2.46×10^{-5}
ZDT2	1.04×10^{-3}	4.05×10^{-3}	1.28×10^{-4}	9.01×10^{-5}	8.95×10^{-5}	7.56×10^{-5}	4.62×10^{-5}	2.18×10^{-6}
ZDT3	1.77×10^{-4}	6.11×10^{-5}	1.76×10^{-4}	4.73×10^{-5}	1.24×10^{-4}	4.32×10^{-5}	1.00×10^{-4}	5.93×10^{-6}
ZDT4	1.17×10^{-1}	7.37×10^{-1}	1.23×10^{-4}	1.09×10^{-4}	7.71×10^{-5}	3.60×10^{-5}	6.98×10^{-5}	7.70×10^{-6}
ZDT6	6.64×10^{-2}	4.67×10^{-2}	6.47×10^{-2}	5.22×10^{-2}	5.66×10^{-2}	5.74×10^{-2}	5.65×10^{-2}	4.64×10^{-2}
DTLZ4	6.33×10^{-3}	1.60×10^{-3}	6.33×10^{-3}	1.82×10^{-3}	6.70×10^{-3}	2.23×10^{-3}	3.29×10^{-3}	1.08×10^{-3}
DTLZ6	8.91×10^{-3}	3.44×10^{-2}	4.32×10^{-3}	2.70×10^{-2}	1.33×10^{-2}	5.35×10^{-2}	5.97×10^{-5}	2.55×10^{-6}
UF1	1.83×10^{-2}	3.07×10^{-2}	1.02×10^{-2}	1.52×10^{-2}	9.98×10^{-3}	1.68×10^{-2}	7.42×10^{-3}	7.85×10^{-3}
UF2	7.40×10^{-3}	1.14×10^{-2}	5.51×10^{-3}	1.06×10^{-2}	7.91×10^{-3}	9.10×10^{-3}	5.39×10^{-3}	9.14×10^{-3}
UF3	1.04×10^{-2}	4.36×10^{-3}	1.01×10^{-2}	4.78×10^{-3}	7.99×10^{-3}	3.61×10^{-3}	7.96×10^{-3}	3.36×10^{-3}
UF4	4.95×10^{-3}	5.54×10^{-4}	4.95×10^{-3}	6.42×10^{-4}	4.93×10^{-3}	3.98×10^{-4}	4.57×10^{-3}	6.39×10^{-4}
UF5	1.09×10^{-1}	9.37×10^{-2}	8.88×10^{-2}	8.88×10^{-2}	8.83×10^{-2}	7.85×10^{-2}	6.46×10^{-2}	4.56×10^{-2}
UF6	2.84×10^{-1}	1.97×10^{-1}	2.50×10^{-1}	1.52×10^{-1}	2.23×10^{-1}	2.19×10^{-1}	1.99×10^{-1}	1.27×10^{-1}
UF7	1.48×10^{-2}	2.33×10^{-2}	1.23×10^{-2}	2.10×10^{-2}	6.89×10^{-3}	7.72×10^{-3}	6.69×10^{-3}	9.77×10^{-3}
UF8	1.32×10^{-1}	1.63×10^{-1}	1.13×10^{-1}	1.14×10^{-1}	1.33×10^{-1}	1.37×10^{-1}	3.08×10^{-1}	1.80×10^{-1}
UF9	1.48×10^{-1}	1.58×10^{-1}	2.04×10^{-1}	2.55×10^{-1}	1.48×10^{-1}	1.53×10^{-1}	2.09×10^{-1}	1.03×10^{-1}
UF10	1.50×10^{0}	1.35×10^{0}	8.58×10^{-1}	8.30×10^{-1}	9.25×10^{-1}	7.33×10^{-1}	1.05×10^{0}	6.16×10^{-1}
Kursawe	5.42×10^{-4}	4.10×10^{-4}	5.30×10^{-4}	2.24×10^{-4}	5.33×10^{-4}	2.56×10^{-4}	1.91×10^{-4}	2.25×10^{-5}
Viennet2	1.50×10^{-3}	7.44×10^{-4}	1.56×10^{-3}	9.64×10^{-4}	1.21×10^{-3}	5.80×10^{-4}	8.50×10^{-4}	3.45×10^{-4}
Viennet3	6.28×10^{-4}	7.31×10^{-4}	6.54×10^{-4}	5.53×10^{-4}	5.70×10^{-4}	4.00×10^{-4}	2.31×10^{-4}	8.05×10^{-5}

Note: The underline indicates the best performance.

Table 9. Results of HV metric on benchmark functions.

Fun	MOAVOA		SMOAVOA		JMOAVOA		IMOAVOA	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
ZDT1	7.14×10^{-1}	1.66×10^{-3}	7.14×10^{-1}	9.43×10^{-4}	7.13×10^{-1}	4.21×10^{-3}	7.20×10^{-1}	2.15×10^{-5}
ZDT2	4.39×10^{-1}	1.94×10^{-3}	4.39×10^{-1}	1.47×10^{-3}	4.39×10^{-1}	3.22×10^{-3}	4.45×10^{-1}	2.22×10^{-5}
ZDT3	5.99×10^{-1}	4.71×10^{-3}	6.00×10^{-1}	5.18×10^{-3}	6.05×10^{-1}	1.77×10^{-2}	6.00×10^{-1}	2.45×10^{-5}
ZDT4	7.13×10^{-1}	1.39×10^{-3}	7.14×10^{-1}	1.27×10^{-3}	7.14×10^{-1}	1.65×10^{-3}	7.20×10^{-1}	2.36×10^{-5}
ZDT6	3.85×10^{-1}	8.39×10^{-4}	3.85×10^{-1}	6.84×10^{-4}	3.85×10^{-1}	5.93×10^{-4}	3.89×10^{-1}	8.06×10^{-5}
DTLZ4	5.07×10^{-1}	1.68×10^{-2}	5.10×10^{-1}	1.57×10^{-2}	5.04×10^{-1}	1.90×10^{-2}	5.35×10^{-1}	3.99×10^{-3}
DTLZ6	1.96×10^{-1}	1.71×10^{-3}	1.96×10^{-1}	1.01×10^{-3}	1.96×10^{-1}	3.05×10^{-3}	2.00×10^{-1}	3.53×10^{-5}
UF1	6.91×10^{-1}	5.79×10^{-3}	6.91×10^{-1}	3.88×10^{-3}	6.96×10^{-1}	3.29×10^{-3}	7.04×10^{-1}	3.42×10^{-3}
UF2	6.98×10^{-1}	4.35×10^{-3}	6.98×10^{-1}	3.92×10^{-3}	7.00×10^{-1}	2.97×10^{-3}	7.13×10^{-1}	1.02×10^{-3}
UF3	4.67×10^{-1}	5.30×10^{-2}	4.55×10^{-1}	4.09×10^{-2}	4.85×10^{-1}	4.88×10^{-2}	5.04×10^{-1}	4.14×10^{-2}
UF4	3.82×10^{-1}	3.75×10^{-3}	3.82×10^{-1}	3.09×10^{-3}	3.80×10^{-1}	3.06×10^{-3}	3.99×10^{-1}	1.88×10^{-3}
UF5	2.06×10^{-1}	6.17×10^{-2}	2.12×10^{-1}	5.68×10^{-2}	2.64×10^{-1}	6.48×10^{-2}	2.57×10^{-1}	5.31×10^{-2}
UF6	1.13×10^{-1}	7.65×10^{-2}	1.19×10^{-1}	7.23×10^{-2}	1.42×10^{-1}	7.94×10^{-2}	1.64×10^{-1}	8.30×10^{-2}
UF7	5.57×10^{-1}	4.84×10^{-3}	5.58×10^{-1}	3.73×10^{-3}	5.59×10^{-1}	3.53×10^{-3}	5.69×10^{-1}	3.22×10^{-3}
UF8	4.22×10^{-1}	2.54×10^{-2}	4.13×10^{-1}	2.79×10^{-2}	4.15×10^{-1}	2.93×10^{-2}	4.34×10^{-1}	3.21×10^{-2}
UF9	6.77×10^{-1}	2.85×10^{-2}	6.78×10^{-1}	2.58×10^{-2}	6.77×10^{-1}	3.29×10^{-2}	6.90×10^{-1}	3.35×10^{-2}
UF10	1.85×10^{-1}	6.78×10^{-2}	2.45×10^{-1}	6.58×10^{-2}	2.22×10^{-1}	8.72×10^{-2}	2.47×10^{-1}	8.57×10^{-2}
Kursawe	5.19×10^{-1}	2.82×10^{-2}	5.34×10^{-1}	3.45×10^{-2}	5.19×10^{-1}	2.56×10^{-2}	5.04×10^{-1}	3.36×10^{-4}
Viennet2	3.38×10^{-1}	4.43×10^{-4}	3.38×10^{-1}	4.26×10^{-4}	3.38×10^{-1}	4.43×10^{-4}	3.38×10^{-1}	2.17×10^{-4}
Viennet3	1.82×10^{-1}	5.10×10^{-4}	1.82×10^{-1}	4.80×10^{-4}	1.82×10^{-1}	4.06×10^{-4}	1.83×10^{-1}	8.34×10^{-5}

Note: The underline indicates the best performance.

4.5. Simulation Experiment on Optimization of Aluminum Smelting Process Parameters

To further analyze the effectiveness of the proposed method in the parameters optimization of aluminum smelting process, IMOAVOA, MOAOA, MOALO, and MOPSO are used to solve the multi-objective problem of the aluminum smelting process described by Equation (1) in Section 2. The algorithm parameters were set according to Table 3. The optimization of the multi-objective problem for the aluminum smelting process is performed every 12 min to update the fuel gas flow rate, combustion air flow rate, air–fuel ratio. One of the optimization results for the multi-objective problem of the aluminum smelting process is shown in Figure 4. Figure 4 indicates that increasing the accuracy of temperature regulation may result in higher energy consumption. However, excessive fuel gas savings can lead to furnace temperatures that are 5 °C or more below set point,

which in turn negatively affects the effectiveness of aluminum smelting. To some extent, the amount of fuel gas used is inversely related to the temperature deviation.

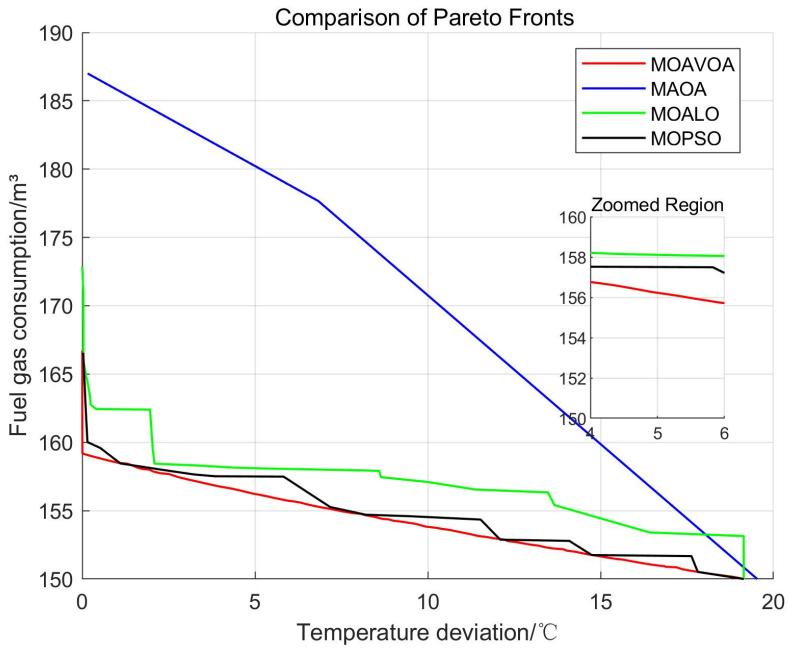


Figure 4. Optimization results of aluminum smelting process setting parameters.

When selecting the optimal setting parameters for aluminum production, it is important to consider both the temperature deviation and the energy consumption. Simultaneously, it is important to ensure accurate furnace temperature regulation while minimizing energy consumption. Based on expert experience, the optimal setting parameters for actual production processes should result in a temperature deviation of less than ±5 °C and the least amount of energy consumption. Therefore, we choose the solution with temperature deviation within ±5 °C and minimum energy consumption as the optimal parameter, based on the above trade-offs.

The results in Figure 4 show that IMOAVOA and MOPSO are the most effective for the optimal design of the aluminum smelting process; the optimal parameters under these two methods are selected to compare with the results from manual adjustments parameters. The data in Figure 5 is derived from the energy consumption generated under manual adjustments for the seven days of the regenerative aluminum plant statistics, as well as a comparison of the fuel gas consumption data generated using MOPSO-based adjustments and IMOAVOA-based adjustments for the same amount of time in operation. The total running time for each day is the time that the regenerative aluminum smelting furnace was statistically running at 800 °C operating conditions. Samples were taken every 12 min, with 10 h of operation at this condition counted on the first day, 9 h on the second day, 7-and-a-half hours on the third and fifth days, and 8 h on the fourth, sixth and seventh days. Analyzing the data in Figure 5 shows that the parameters optimized with IMOAVOA can consume about 1% less fuel gas per week than the parameters optimized with MOPSO in the aluminum smelting process, which is a reduction of 729 m^3 of fuel gas consumption compared to the manual adjustment, 7.67% of the weekly consumption. This reduction is very important both from an environmental and an economic point of view. Table 10 shows a set of optimal parameters and optimization results obtained by IMOAVOA solving the multi-objective problem for the aluminum smelt process presented in Section 2.2. The fuel gas consumption in Table 10 is statistically generated for 12 min at 800 °C operating conditions.

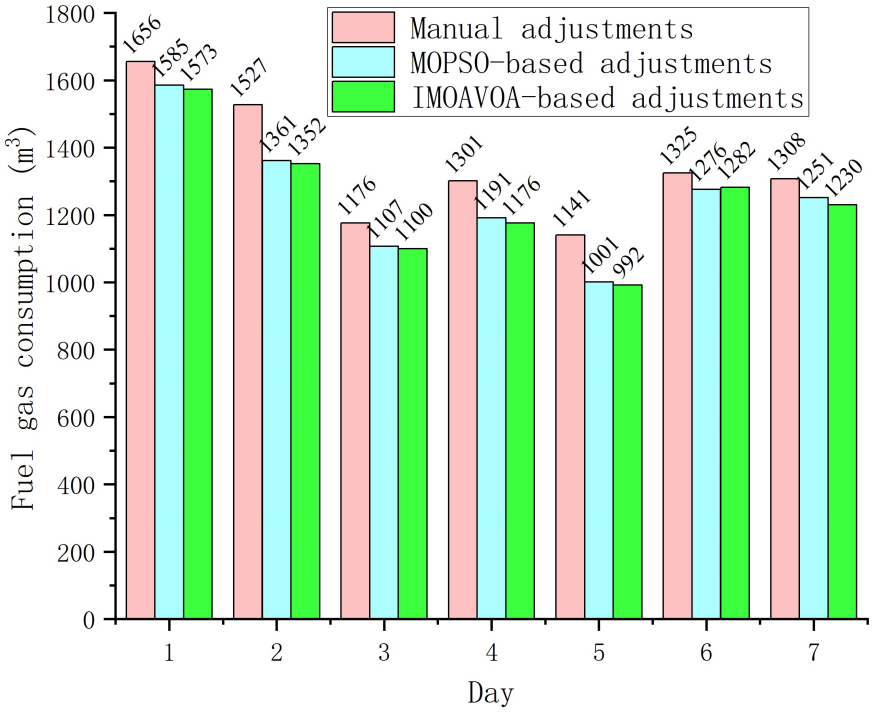


Figure 5. Fuel gas consumption of the aluminum smelting process with different settings of parameters.

Table 10. A set of optimal parameters and optimization results of the aluminum smelting process.

Aluminum Smelting Process Parameters	Solutions		Results
fuel gas flow rate	131.47 (m^3/h)	fuel gas consumption	26.29 m^3
Air–fuel gas ratio	3.42		
Combustion air flow rate	449.6 (m^3/h)	Temperature deviation	4.7 °C

4.6. Industrial Experiments

In order to verify the effectiveness of the proposed energy optimization design for the aluminum smelting process, a one-day experiment was conducted in a recycled aluminum smelting plant. The experiment was divided into three groups, one for manual adjustments, one for MOPSO optimization adjustments and one for IMOAVOA optimization adjustments. The experiments were carried out at a furnace temperature set at 800 °C. Due to the limitation of the device, the optimization adjustments using the algorithm are discontinuous and a static optimization, where the optimization adjustments are carried out every 12 min and the fuel gas consumption is counted for 10 h under this operating condition. The temperature variations under the three types of adjustments are shown in Figure 6. It can be seen that both the optimized adjustments by IMOAVOA and the optimized adjustments by MOPSO are more stable than the manual adjustments in terms of temperature change, but due to the interference of the random factors and the hysteresis of the static optimization, it is difficult to maintain the temperature at the set temperature of ±5 °C. Figure 7 shows the comparison of the energy consumption of the smelting furnace under the three adjustments for ten hours of operation at 800 °C. A total of 43 m^3 less fuel gas is consumed under IMOAVOA adjustments compared to manual adjustments, and 14 m^3 less than MOPSO optimization adjustments. Analysis of the results of the experiments shows that the use of IMOAVOA optimization adjustments for the aluminum smelting parameters can effectively reduce the consumption of gas. The optimization of aluminum smelting parameters using IMOAVOA can effectively reduce gas consumption.

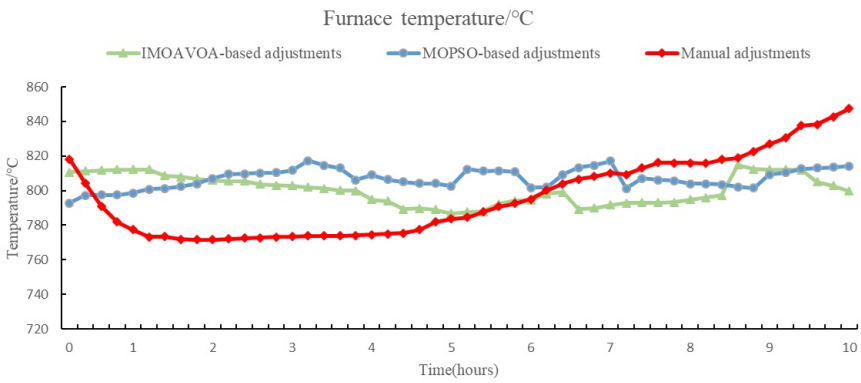


Figure 6. Variation in furnace temperature in regenerative smelt with different types of adjustments.

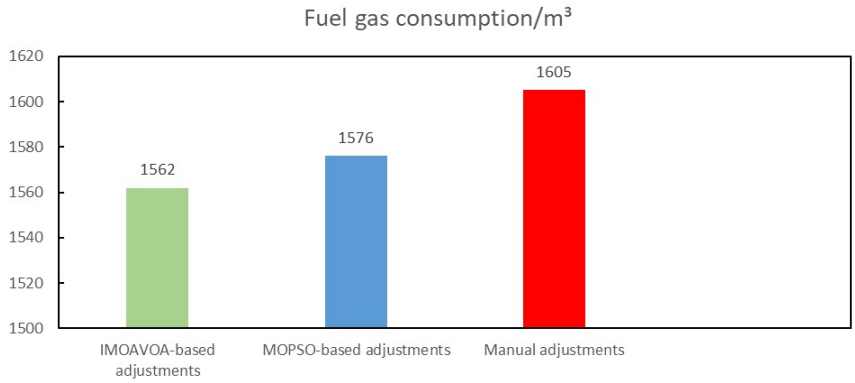


Figure 7. Comparison of fuel gas consumption for different types of adjustments.

5. Conclusions and Future Directions

In this paper, an improved multi-strategy multi-objective artificial vulture optimization algorithm (IMOAVOA) is proposed. The IMOAVOA includes a multi-directed leader selection strategy and retains a portion of the grid mechanism to select leaders that intensify the exploration potential of the IMOAVOA, while obtaining more solutions with high diversity. Additionally, a novel perturbation bounce strategy is developed to improve the ability to find quality solutions of the IMOAVOA. A dynamic switching–elimination mechanism based on crowding distance (DSECD) is proposed to maintain the archive, which enhances the convergence of the IMOAVOA, the coverage of Pareto frontiers, and the diversity of solutions.

The performance of the IMOAVOA is evaluated using multi-objective algorithms, including MOPSO, MOALO and MOAOA, on 20 test functions. The experimental results demonstrate that the IMOAVOA performs well on the twenty test functions by observing the IGD, GD, and HV metrics. The results indicate that IMOAVOA outperforms the three compared algorithms in terms of algorithm convergence, the Pareto front coverage, and the solution diversity. The stepwise improvement comparison experiments demonstrate the effectiveness of individual improvements, each of which enhances the performance of the IMOAVOA to a different degree. Notably, IMOAVOA performs quite well on optimization problems with two optimization objectives. IMOAVOA has proved to be more suitable than other traditional algorithms for the optimal design of aluminum smelting processes. This is essential for improving product quality and the economic efficiency of the aluminum smelting process. However, there are limitations in using IMOAVOA for the optimal design of energy consumption for aluminum smelting process. The optimization effect of the algorithm is affected by some random factors in the plant. The limitations of the aluminum smelting instruments lead to the fact that performing real time optimization is difficult

and only static optimization can be performed. IMOAVOA can be applied to optimization problems, such as strength and weight optimization in mechanical design, loss and stability optimization in electric power systems, yield and energy consumption optimization in chemical engineering, and time and cost optimization in transportation planning.

Author Contributions: Conceptualization, J.D. and H.L.; methodology, J.D. and H.L.; software, H.L. and Y.Z.; validation, H.L., H.S. and P.L.; formal analysis and investigation, H.L., Y.Z. and P.L.; data curation, H.L. and H.S.; writing—original draft preparation, J.D. and H.L; writing—review and editing, J.D. and H.L; visualization, H.L. and Y.Z.; supervision, J.D. and H.L.; project administration, J.D.; funding acquisition, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Nos.62341302, 62273111), and the Guangxi Natural Science Foundation (No.2024JJA170205).

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality requested by the enterprise providing the data.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Mostaghim, S.; Teich, J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In Proceedings of the 2003 IEEE Swarm Intelligence Symposium, Indianapolis, IN, USA, 24–26 April 2003; SIS'03 (Cat. No. 03EX706); pp. 26–33.
2. Mirjalili, S.; Jangir, P.; Saremi, S. Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Appl. Intell.* **2017**, *46*, 79–95. [CrossRef]
3. Mirjalili, S.; Jangir, P.; Mirjalili, S.Z.; Saremi, S.; Trivedi, I.N. Optimization of problems with multiple objectives using the multi-verse optimization algorithm. *Knowl.-Based Syst.* **2017**, *134*, 50–71. [CrossRef]
4. Khodadadi, N.; Abualigah, L.; El-Kenawy, E.S.M.; Snasel, V.; Mirjalili, S. An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems. *IEEE Access* **2022**, *10*, 106673–106698. [CrossRef]
5. Khodadadi, N.; Soleimanian Gharehchopogh, F.; Mirjalili, S. MOAVOA: A new multi-objective artificial vultures optimization algorithm. *Neural Comput. Appl.* **2022**, *34*, 20791–20829. [CrossRef]
6. Zhao, W.; Zhang, Z.; Mirjalili, S.; Wang, L.; Khodadadi, N.; Mirjalili, S.M. An effective multi-objective artificial hummingbird algorithm with dynamic elimination-based crowding distance for solving engineering design problems. *Comput. Methods Appl. Mech. Eng.* **2022**, *398*, 115223. [CrossRef]
7. Hassan, M.H.; Kamel, S.; Abualigah, L.; Eid, A. Development and application of slime mould algorithm for optimal economic emission dispatch. *Expert Syst. Appl.* **2021**, *182*, 115205. [CrossRef]
8. Dhiman, G.; Singh, K.K.; Soni, M.; Nagar, A.; Dehghani, M.; Slowik, A.; Kaur, A.; Sharma, A.; Houssein, E.H.; Cengiz, K. MOSOA: A new multi-objective seagull optimization algorithm. *Expert Syst. Appl.* **2021**, *167*, 114150. [CrossRef]
9. bin Mohd Zain, M.Z.; Kanesan, J.; Chuah, J.H.; Dhanapal, S.; Kendall, G. A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization. *Appl. Soft Comput.* **2018**, *70*, 680–700. [CrossRef]
10. Dong, J.; Dou, Z.; Si, S.; Wang, Z.; Liu, L. Optimization of capacity configuration of wind–solar–diesel–storage using improved sparrow search algorithm. *J. Electr. Eng. Technol.* **2022**, *17*, 1–14. [CrossRef]
11. Zouache, D.; Got, A.; Drias, H. An external archive guided Harris Hawks optimization using strengthened dominance relation for multi-objective optimization problems. *Artif. Intell. Rev.* **2023**, *56*, 2607–2638. [CrossRef]
12. Kahraman, H.T.; Akbel, M.; Duman, S.; Kati, M.; Sayan, H.H. Unified space approach-based Dynamic Switched Crowding (DSC): A new method for designing Pareto-based multi/many-objective algorithms. *Swarm Evol. Comput.* **2022**, *75*, 101196. [CrossRef]
13. Tian, Z.; Wang, J. Variable frequency wind speed trend prediction system based on combined neural network and improved multi-objective optimization algorithm. *Energy* **2022**, *254*, 124249. [CrossRef]
14. Hinojosa, S.; Oliva, D.; Cuevas, E.; Pajares, G.; Avalos, O.; Gálvez, J. Improving multi-criterion optimization with chaos: A novel Multi-Objective Chaotic Crow Search Algorithm. *Neural Comput. Appl.* **2018**, *29*, 319–335. [CrossRef]
15. Qiu, R.; Wang, D.; Singh, V.P.; Wang, Y.; Wu, J. Integration of deep learning and improved multi-objective algorithm to optimize reservoir operation for balancing human and downstream ecological needs. *Water Res.* **2024**, *253*, 121314. [CrossRef]
16. Wang, X.l.; Lu, M.y.; Wei, S.m.; Xie, Y.f. Multi-objective optimization based optimal setting control for industrial double-stream alumina digestion process. *J. Cent. South Univ.* **2022**, *29*, 173–185. [CrossRef]

17. Dai, H.; Zhao, J.; Wang, Z.; Chen, C.; Liu, X.; Guo, Z.; Chen, Y.; Zhang, S.; Li, J.; Geng, H.; et al. Optimal control of sewage treatment process using a dynamic multi-objective particle swarm optimization based on crowding distance. *J. Environ. Chem. Eng.* **2023**, *11*, 109484. [CrossRef]
18. Xie, Y.; Du, L.; Zhao, J.; Liu, C.; Li, W. Multi-objective optimization of process parameters in stamping based on an improved RBM–BPNN network and MOPSO algorithm. *Struct. Multidiscip. Optim.* **2021**, *64*, 4209–4235. [CrossRef]
19. Kalita, K.; Chohan, J.S.; Jangir, P.; Chakraborty, S. A new decomposition-based multi-objective symbiotic organism search algorithm for solving truss optimization problems. *Decis. Anal. J.* **2024**, *10*, 100371. [CrossRef]
20. Zhang, H.; Ma, L.; Wang, J.; Wang, L. Furnace-grouping problem modeling and multi-objective optimization for special aluminum. *IEEE Trans. Emerg. Top. Comput. Intell.* **2021**, *6*, 544–555. [CrossRef]
21. Li, L.L.; Ji, B.X.; Lim, M.K.; Tseng, M.L. Active distribution network operational optimization problem: A multi-objective tuna swarm optimization model. *Appl. Soft Comput.* **2024**, *150*, 111087. [CrossRef]
22. Li, Y.; Wang, Z.; Cheng, Y.; Tang, Y.; Shang, Z. A hybrid improved moth-flame optimization with differential evolution with global and local neighborhoods algorithm for pose optimization on a space manipulator. *Meas. Sci. Technol.* **2019**, *30*, 125012. [CrossRef]
23. Hu, K.; Mo, Y. A novel unmanned aerial vehicle path planning approach: Sand cat optimization algorithm incorporating learned behaviour. *Meas. Sci. Technol.* **2024**, *35*, 046203. [CrossRef]
24. Yankai, W.; Shilong, W.; Dong, L.; Chunfeng, S.; Bo, Y. An improved multi-objective whale optimization algorithm for the hybrid flow shop scheduling problem considering device dynamic reconfiguration processes. *Expert Syst. Appl.* **2021**, *174*, 114793. [CrossRef]
25. Wang, X.; Zhang, X.; Qin, B.; Guo, L. Improved multi-objective grasshopper optimization algorithm and application in capacity configuration of urban rail hybrid energy storage systems. *J. Energy Storage* **2023**, *72*, 108363. [CrossRef]
26. Nematollahi, M.; Ghaffari, A.; Mirzaei, A. Task offloading in Internet of Things based on the improved multi-objective aquila optimizer. *Signal Image Video Process.* **2024**, *18*, 545–552. [CrossRef]
27. Lu, C.; Gao, L.; Li, X.; Xiao, S. A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. *Eng. Appl. Artif. Intell.* **2017**, *57*, 61–79. [CrossRef]
28. Fu, C.; Lin, K.P.; Zhou, Y.T. Optimal capacity configuration of CCHP system with improved operation strategies using improved multi-objective multi-universe algorithm. *Expert Syst. Appl.* **2022**, *199*, 117183. [CrossRef]
29. Luo, Y.; Dai, J.; Chen, X.; Liu, Y. A Hybrid Modeling Method for Aluminum Smelting Process Based on a Hybrid Strategy-Based Sparrow Search Algorithm. *IEEE Access* **2022**, *10*, 101149–101159. [CrossRef]
30. Zhang, Y.; Dai, J.; Huang, Q.; Luo, Y. An Improved Sand Cat Swarm Optimization and Its Application in Aluminum Melting Process. In Proceedings of the 2023 5th International Conference on Intelligent Control, Measurement and Signal Processing (ICMSP), Chengdu, China, 19–21 May 2023; pp. 380–384.
31. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [CrossRef]
32. Zhang, Q.; Zhou, A.; Zhao, S.; Suganthan, P.N.; Liu, W.; Tiwari, S. Multiobjective optimization test instances for the CEC 2009 special session and competition. *Mech. Eng.* **2008**, *264*, 1–30.
33. Lim, W.J.; Jambek, A.B.; Neoh, S.C. Kursawe and ZDT functions optimization using hybrid micro genetic algorithm (HMGA). *Soft Comput.* **2015**, *19*, 3571–3580. [CrossRef]
34. Gong, Y.; Wang, W.; Gong, S. A Novel Self-Adaptive Mixed-Variable Multiobjective Ant Colony Optimization Algorithm in Mobile Edge Computing. *Secur. Commun. Netw.* **2022**, *2022*, 4967775. [CrossRef]
35. Chase, N.; Rademacher, M.; Goodman, E.; Averill, R.; Sidhu, R. A benchmark study of multi-objective optimization methods. *BMK-3021 Rev.* **2009**, *6*, 1–24.
36. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable multi-objective optimization test problems. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; CEC'02 (Cat. No. 02TH8600); Volume 1, pp. 825–830.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.