Journal of Artificial Intelligence and Control Systems

CoSci Press

Research Article

Multi-objective Educational Competition Algorithm and Its Engineering Application

Weidong Cao [1,2,*], Guorong Wang [1], Tian Pan [1], Yuanqi Jia [1], Jianjun Ni [1]

1 College of Artificial Intelligence and Automation, Hohai University, Changzhou 213200, China

2 Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University, Changzhou 213200, China

Article History:

Received: 22 June 2025

Revised: 31 July 2025

Accepted: 14 August 2025

Published: 23 September 2025

Abstract: Educational Competition Optimization (ECO), as a new metaheuristic algorithm, performs well in single-objective optimization problems, but it is difficult to apply to multi-objective complex optimization problems. Hence, a Multi-Objective Educational Competition Optimization (MOECO) based on the Pareto improvement and the roulette strategy is proposed. In order to verify the effect of MOECO in engineering application, MOECO was used to solve the hobbing control parameter optimization model. Compared with the Multi-Objective Gray Wolf Algorithm (MOGWO) and the Multi-Objective Runge Kutta Optimizer (MORUN), the experimental results show that MOECO is feasible and effective in this engineering case. MOECO provides a new solution for complex multi-objective optimization problems, which has theoretical value and engineering application potential.

Keywords: meta-heuristic algorithm; MOECO; hobbing control parameter optimization

1. Introduction

In recent years, with the rapid development of artificial intelligence technology, novel metaheuristic algorithms have emerged as powerful tools for efficiently solving various problems, owing to their outstanding global search and optimization capabilities, as well as their ability to quickly handle complex, nonlinear problems [1]. Currently, mainstream metaheuristic algorithms include the Grey Wolf Optimizer (GWO) [2], Harris Hawks Optimization (HHO) [3], Sparrow Search Algorithm (SSA) [4], and others. These metaheuristic algorithms are highly versatile, exhibit high search efficiency, are straightforward to implement, and are not restricted by gradient information, with each possessing unique strengths and application scenarios.

The single-objective meta-heuristic algorithm can no longer satisfy the complex scenarios of the current industrial environment, and it is also difficult to apply to some high-dimensional nonlinear problems. The multi-objective meta-heuristic algorithm came into being. Multi-objective metaheuristic algorithms aim to solve optimization problems with multiple conflicting objectives by balancing trade-offs between different goals to identify a set of equilibrium solutions (Pareto optimal solutions). These include algorithms such as multi-objective genetic algorithms [5], multi-objective spotted hyena optimizer [6], and others. They are well-suited for scenarios involving high-dimensional objectives, discontinuous solution spaces, or problems where explicit mathematical modeling is challenging, such as engineering optimization, resource allocation, and scheduling.

Multi-objective genetic algorithms are heuristic algorithms developed based on genetic algorithms, simulating biological evolution through selection, crossover, and mutation to handle multiple objective functions simultaneously in a single optimization process. In highway construction optimization [7], they delivered efficient

[*] Corresponding author: Weidong Cao, College of Artificial Intelligence and Automation, Hohai University, Changzhou 213200, China; Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, Hohai University, Changzhou 213200, China, cwd2018@hhu.edu.cn

https://doi.org/10.xxxx/xxxxx

solutions by balancing conflicting objectives such as time, cost, quality, and environmental impact (e.g., carbon emissions [8]). In earthwork engineering [9], they optimized excavation volume, transportation costs, and fuel consumption, thereby enhancing both project efficiency and sustainability. Furthermore, in highway construction [10], they facilitated trade-offs between greenhouse gas emissions and project costs [11], promoting green construction practices. Multi-objective particle swarm optimization (MOPSO) dynamically maintained a non-dominated solution archive to guide the swarm toward the Pareto front, often incorporating crowding distance or clustering strategies to preserve diversity. In construction project management [12], it optimized time, cost, and quality while using fuzzy logic to handle uncertain constraints. It also aided in the co-optimization of project costs and CO_2 emissions, thereby quantifying conflicts between environmental and economic objectives [13]. The multi-objective spotted hyena optimizer (MOSHO) [14] was applied to multi-objective optimization problems via simulating the social and hunting behaviors of spotted hyenas [15]. Tests on 24 benchmark functions demonstrated its strong convergence and ability to generate high-quality Pareto optimal solutions. The multi-objective manta ray foraging optimization algorithm (MOMRFO) [16] struck an effective balance between the diversity and convergence of the Pareto set.

In 2024, the Educational Competition Optimization (ECO) algorithm was proposed [17]. It is a meta-heuristic optimization algorithm inspired by the competition and cooperation mechanisms in educational systems. It simulates the process of students improving their abilities through competition, learning and interaction in the educational process, analogizes the "knowledge level" to the solution of optimization problems, and seeks the optimal solution through iterative updates. However, like other traditional single-objective metaheuristic algorithms, ECO also fails to adapt to the complex scenarios of the current industrial environment and is hardly applicable to certain high-dimensional nonlinear problems. Hence, its improvement and application have become particularly significant. After improvement, traditional metaheuristic algorithms have enhanced applicability and intelligence across multiple fields, with their scope of application extended to handling multi-objective conflicts, demonstrating excellent robustness and convergence. To address ECO's incapability of solving multi-objective optimization problems, Pareto improvement and roulette strategy are integrated into the ECO algorithm, forming the Multi-objective Educational Competition Optimization (MOECO) algorithm. By extending this single-objective metaheuristic algorithm to handle multiple objectives, not only can its inherent limitations be effectively mitigated, but the algorithmic repertoire for solving engineering problems is also enriched. This provides users with additional options, thereby holding profound significance for practical engineering applications. The general process is as follows. First, the population is initialized by randomly assigning values to the positions of schools and students. Then, the iterative loop is entered: the objective function values of individuals are computed, and the global elite solutions are updated; the Pareto improvement strategy is used to screen non-dominated solutions, and the archive set is updated; several elite solutions are selected from the archive set using the roulette wheel strategy, and the central position of these elite solutions is calculated; the positions of schools and students are updated in stages according to the primary, junior high, and senior high school update strategies of ECO. Finally, the loop is continuously executed until the termination condition is met; otherwise, the algorithm is terminated, and a set of non-dominated solutions is obtained.

This paper is organized as follows. Section 2 introduces the proposed methods, including the problem description and specific process. Engineering tests were conducted in Section 3. Section 4 presents the conclusion.

2. Methods

2.1. Problem Description

A multi-objective optimization problem (MOP), as the name suggests, essentially involves optimizing multiple conflicting objective functions simultaneously. A classic MOP [2] can be mathematically expressed as:

$$\begin{aligned} &\min F(X) = \left(f_1(X), f_2(X), \cdots, f_m(X)\right)^T \\ &s.t. \quad g_j(X) \le 0,\ j = 1,2,\cdots,p \\ &\qquad h_k(X) = 0,\ k = 1,2,\cdots,q \\ &\qquad x_i \in X,\ L_i \le x_i \le U_i,\ i = 1,2,\cdots,n \end{aligned} \tag{1}$$

where $F(X)$ is a vector composed of m objective functions, n is the number of variables, p is the number of inequality constraints, q is the number of equality constraints, $g_j(X)$ represents the i-th inequality constraint, $h_k(X)$ denotes the k-th equality constraint, and $[L_i, U_i]$ is the boundary of the i-th variable [18].

Many control problems in reality can be transformed into formulas similar to the above. Previous studies by various scholars have shown that multi-objective metaheuristic algorithms are highly proficient in solving such problems, and

subsequent collaboration with decision-making methods can yield optimal variables that meet user requirements [19]. However, according to the No Free Lunch (NFL) theorem, different algorithms are suited to distinct application scenarios. In light of this, the ECO algorithm has been improved in this work to serve as a multi-objective optimizer for solving the hobbing control parameter optimization problem.

2.2. MOECO

Pareto improvement and roulette strategy are integrated into ECO, forming MOECO. The individual update of ECO is still applied in MOECO. Pareto improvement is mainly used to obtain non-dominated solutions under multiple objectives. Roulette wheel selection is mainly used to select excellent individuals *EliteSet*. For non-dominated solution management, a new *Archive* set and *EliteSet* are added. In each iteration, non-dominated solutions from the current population update *EliteSet*, which is then merged into *Archive*. Non-dominated sorting and crowding distance calculation are applied; if *Archive* exceeds the maximum archive capacity *AMS*, solutions with the lowest crowding distance are removed, dynamically maintaining high-quality solutions and boosting Pareto frontier search. The population update uses a dynamic mechanism based on iteration rounds (Step 8). Combined with the centroid of 5 elite solutions (selected via roulette), it enables both exploration and exploitation. The basic process of MOECO is as follows, with its corresponding flowchart presented in Figure 1.

Step 1: Set the maximum number of iterations *Max_iter*, population size *N*, maximum archive capacity *AMS*, and initialize the iteration counter t=0.

Step 2: Initialize algorithm parameters (archive size and elite solution storage space), then initialize the population.

Step 3: If t<*Max_iter*, go to Step 4; otherwise, go to Step 10.

Step 4: Evaluate objective function values for all individuals and update the global elite solutions.

Step 5: Add current solutions to the archive, perform non-dominated sorting, and update the archive.

Step 6: If the archive exceeds *AMS*, rank and cull solutions, then update archive rankings.

Step 7: Select 5 elite solutions via roulette wheel selection and compute their centroid.

Step 8: Update positions of schools and students in stages using primary, junior high, and senior high school strategies.

Step 9: Handle boundary constraints and update the global elite to the centroid of elite solutions.

Step 10: Output the final non-dominated solution set.

To elaborate on the algorithm in more detail, the pseudo-code is presented as follows.

Input: $F(X)$, *Max_iter*, N, *AMS*, the proportion of primary school $G1 = 0.2$, *G1Number*=round($N\times G1$), the proportion of middle school $G2 = 0.1$, *G2Number* = round($N\times G2$)

1: Initialize the setting parameters (*Max_iter*, N, *AMS*, etc.)
2: Initialize the archive set *Archive* and elite solution storage *EliteSet*
3: Randomly initialize the population $P = \{ X_1, X_2, ..., X_N \}$
4: for $t = 1$ to *Max_iter* do
5: for each individual $X_j \in P$ do
6: Compute objective values $F(X_j)$, $j=1,2,...,N$
7: end for
8: Update *EliteSet* with non-dominated solutions from P
9: Merge *EliteSet* into *Archive*
10: Apply non-dominated sorting to *Archive*
11: if *SizeCal* > *AMS* (*SizeCal* denotes the number of non-dominated solutions in *Archive*.) then
12: Compute crowding distance for each solution in *Archive*
13: Remove solutions with lowest crowding distance
14: end if
15: Select 5 elite solutions $S=\{s_1, s_2, s_3, s_4, s_5\}$ from *Archive* using roulette wheel selection
16: Compute the centroid c of selected elite solutions
17: Compute dynamic parameters:
18: $pi = 4 \cdot ra \cdot \left(1 - \frac{t}{Max\_iter}\right)$. ra is a random variable following the standard normal distribution
19: $E = \frac{\pi t}{pi \cdot Max\_iter}$
20: $w = 0.1 \cdot \ln\left(2 - \frac{t}{Max\_iter}\right)$
21: for $j = 1$ to N do
22: if mod(t, 3) = 1 then
23: if $j \le G1Number$ then
24: Update school position using Eq. (2)
25: else
26: Update student position using Eq. (3)
27: end if
28: else if mod(t, 3) = 2 then
29: if $j \le G2Number$ then
30: Update school position using Eq. (4)
31: else
32: Update student position using Eq. (5)
33: end if
34: else

35: if $j \leq G2Number$ then
36: Update school position using Eq. (6)
37: else
38: Update student position using Eq. (7)
39: end if
40: end if
41: Apply boundary handling
42: end for
43: Update *EliteSet* = *c*
44: end for
45: Output non-dominated solution set *Archive*

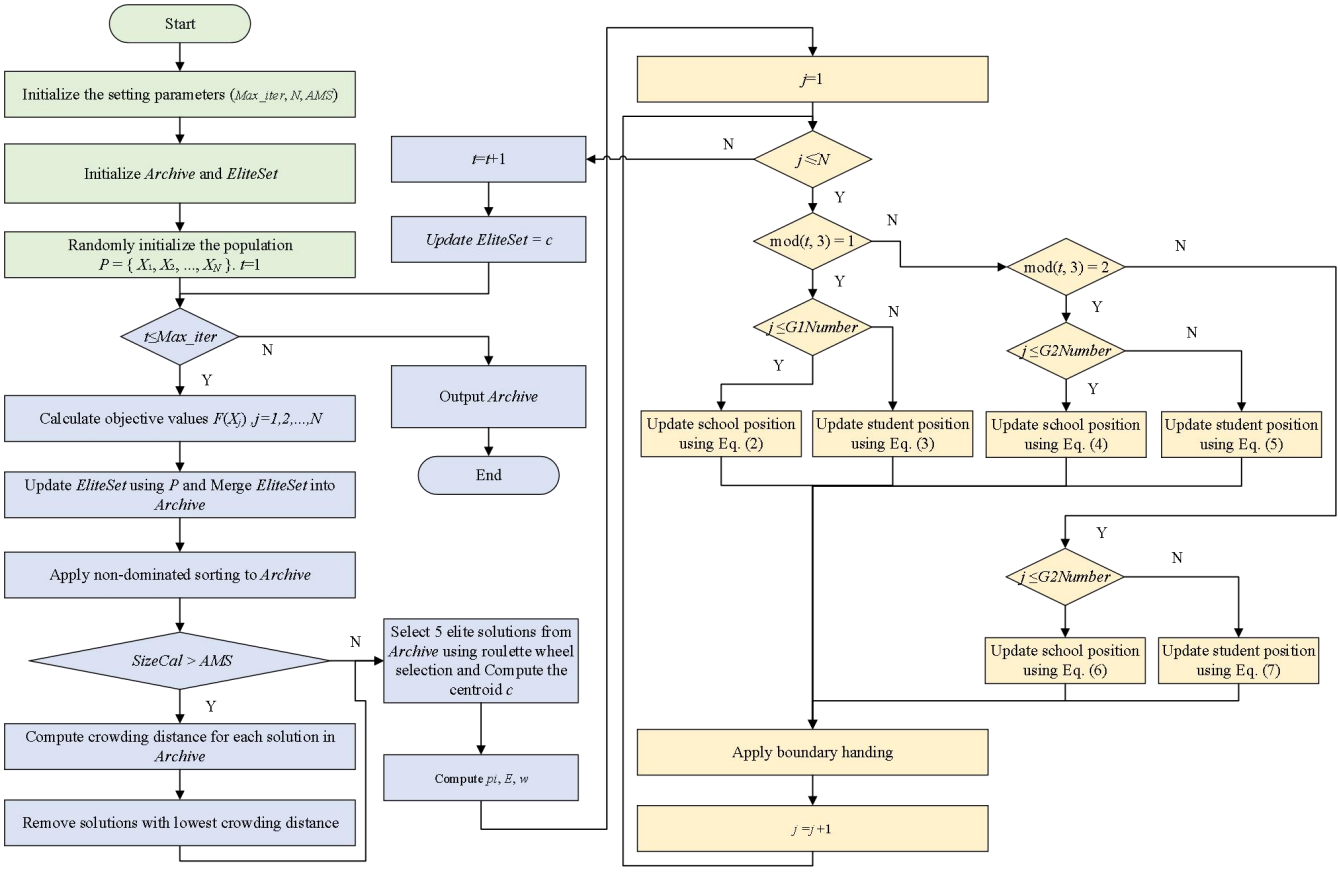

Figure 1. Folwchart of MOECO

Eqs. (2-7) show below.

$$X_j^{new} = X_j + w\left(\text{avg}(P) - X_j\right) * \text{Levy}(n) \tag{2}$$

where X_j^{new} is the next generation value of X_j, avg represents the average position calculation function, * denotes dot multiplication, Levy is a Lévy flight function [17].

$$X_j^{new} = X_j + w\left(\text{close}(X_j) - X_j\right) * randn \tag{3}$$

where close represents the position calculation function closest to school, *randn* is a random number that conforms to the standard normal distribution (i.e., a normal distribution with a mean of 0 and a standard deviation of 1).

$$X_j^{new} = X_j + \exp\left(\frac{t}{Max\_iter} - 1\right)\left(X_{bs} - \text{avg}(P)\right) * \text{Levy}(n) \tag{4}$$

where X_{bs} can be randomly selected from *EliteSet*.

$$X_j^{new} = \begin{cases} X_j - w \cdot \text{close}(X_j) - Q\left(E \cdot w \cdot \text{close}(X_j) - X_j\right), Ra < 0.5 \\ X_j - w \cdot \text{close}(X_j) - Q\left(w \cdot \text{close}(X_j) - X_j\right), Ra \geq 0.5 \end{cases} \tag{5}$$
$$Q = 4 \cdot randn \cdot \left(1 - \frac{j}{Max\_iter}\right)$$

where *Ra* is a random value within 0 and 1.

$$X_j^{new} = X_j + \left(\eta_1 - \eta_2\right) \cdot \left(X_{bs} - X_j\right) \tag{6}$$

where η_1 and η_2 are random numbers that conform to the

standard normal distribution.

$$X_j^{new} = \begin{cases} X_{bs} - Q\left(E \cdot X_{bs} - X_j\right), Rb < 0.5 \\ X_{bs} - Q\left(X_{bs} - X_j\right), Rb \geq 0.5 \end{cases} \quad (7)$$

where Rb is a random value within 0 and 1.

Furthermore, an explanation of the calculation of crowding distance and the centroid c is listed below.

The crowding distance is an important index to measure the density of solutions in the target spatial distribution. It is used to further rank the solutions of the same non-dominated level according to the congestion degree.

$$d_i = \sum_{m=1}^{M} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{max} - f_m^{min}} \quad (8)$$

where M is the number of objective functions, f_m^{i+1} denotes the value of the next solution adjacent to solution i on objective m, while f_m^{i-1} denotes the value of the previous solution adjacent to solution i on objective m. f_m^{max} is the maximum value on objective m, and f_m^{min} is the minimum value on objective m. It should be noted that for boundary points, i.e., the first and last points, the crowding distance should be set to infinity.

The centroid c refers to calculating the average value of the corresponding attribute values of the 5 selected non-dominated solutions, which is similar to the element-wise division operation in MATLAB.

3. Experimental Results

Dry hobbing is an efficient and environmentally friendly gear processing technology [20]. It is an advanced green manufacturing technology without cutting fluid, However, dry hobbing has high requirements for processing conditions, and the requirements for parameters in all aspects are correspondingly high. Its control parameters (such as spindle speed, axial feed, etc.) have complex effects on energy consumption, cutting quality and time. Therefore, it is of great significance to build an efficient method to find the appropriate control parameters. In dry hobbing, the decision-making of control parameters using swarm intelligence algorithm strategy has attracted much attention [21]. In this work, MOECO is applied to the actual hobbing process, and the following three objectives need to be optimized at the same time: the energy consumption (E) includes the total energy consumption of the empty cutting, cutting and auxiliary system [18] (unit: kWh), the cutting quality (Q) is the comprehensive accuracy index composed of tooth alignment error and tooth profile error (unit: mm), the total cutting time (T) is the total time (unit: s), and the decision variables are hob diameter (d_{a0}), spindle speed (n), axial feed (f) and the number of hob heads (z_0).

The objective functions for energy consumption (E), machining quality (Q), and total time (T) follow the model [18,19], defined as:

$$\min F(d_{a0}, z_0, n, f) = (\min E, \min Q, \min T) \quad (9)$$

$$E = P_s t_s + (P_s + P_{sc} + \kappa_1 n + \kappa_2 n^2)(t_a + t_c) + \left((1+\varepsilon_1)P_r + \varepsilon_2 P_r^2\right) t_c$$

$$P_r = C K_1 K_2 K_3 m_n{}^X f^Y h^Z \left(\frac{\pi d_{a0} n}{1000}\right)^{1-U} z_1{}^V / d_{a0}$$

$$Q = w_1 f^2 \frac{\sin\alpha}{4 d_{a0}} + w_2 \frac{\pi^2 m_n z_0{}^2 \sin\alpha}{4 z_1 z_k{}^2}$$

$$T = t_s + t_a + t_c$$

$$t_c = \frac{z_1(E_{in} + B + U_{out})}{z_0 n f}, t_a = \frac{L_a z_1}{z_0 n f} + \frac{L_r}{v_r},$$

$$\text{s.t. } d_{a0,\min} \leq d_{a0} \leq d_{a0,\max}$$

$$z_{0,\min} \leq z_0 \leq z_{0,\max}$$

$$n_{\min} \leq n \leq n_{\max}$$

$$f_{\min} \leq f \leq f_{\max}$$

$$0.0312 f^2 / r_a \leq [Ra]$$

where P_s is the standby power. t_s is the standby time. t_a is the empty cutting time. t_c is the cutting time. P_{sc} is the auxiliary system power when the spindle system is running. κ_1 and κ_2 is the no-load operating power factor. L_a is the axial empty cutting stroke of the hob. L_r is the radial empty cutting stroke of the hob. v_r is the radial feed speed. P_r is the waste removal power. C, X, Y, Z, U and V are the cutting force coefficients. K_1 is the work-piece material correction coefficient. K_2 is the work-piece hardness correction coefficient. K_3 is the work-piece helix angle correction coefficient. ε_1, ε_2 is the additional loss power factor. E_{in} is the cut-in stroke. U_{out} is the cut-out stroke. w_1 and w_2 are the distribution weight of tooth direction error and tooth profile error. z_k is the number of gashes. r_a is the arc radius of tool tip. $[Ra]$ is the limit value of roughness.

The basic experimental conditions are as follows: (1) YE3120CNC7 is equipped with Siemens 840Dsl CNC system. (2) P65 Klingelnberg gear measuring machine, and (3) machine tool multi-source energy consumption status information online detection system [22]. The setup of the environment can be found in Reference 21.

The gear type is involute cylindrical gear (including some non-standard gears), the material is 20CrMo, and the reverse hob feed mode is adopted for processing. The hob base material is S390 powder and metallurgical material, and

the coating is TiAlN. The optimization problem and hobbing model can be found in Reference 21. The range of d_{a0} is [80, 85], the range of z_0 is [1, 2], the range of n is [660, 720], and the range of f is [1.2, 2.6]. MOECO is used to solve the control parameter model. The optimized control parameters under different objectives are obtained, that is, multiple sets of non-dominated solutions. MOECO has obtained 55 sets of non-dominated solutions, and the corresponding Pareto front is shown in Figure 1.

It can be seen from Figure 2 that when Q decreases, the values of E and T will increase, which is also consistent with the actual situation. Improve processing quality at the cost of increasing energy consumption and time. It also shows that there is no really optimal control parameter that can optimize all objectives.

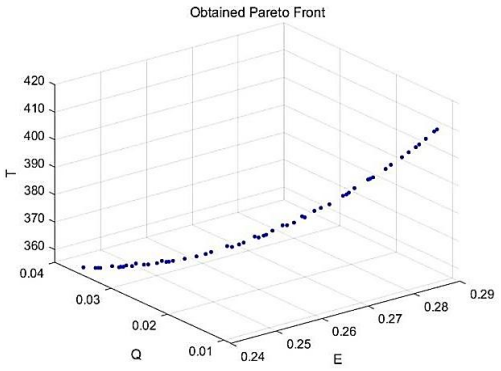

Figure 2. Corresponding Pareto front for optimizing control parameters

4. Discussion

In order to further illustrate the search ability and strong convergence, efficiency and stability of MOECO, comparative experiments were also carried out. Other algorithms for comparison include MOGWO (multi-objective Gray Wolf algorithm) and MORUN (multi-objective Runge Kutta algorithm) [18]. The engineering application principle of the other two algorithms is the same as that of MOECO, and the initial value of control parameters is also the same.

Table 1. Avg and Std data of optimal control parameters for each method

	MOECO	MOGWO	MORUN
	Avg±Std	Avg±Std	Avg±Std
d_{a0}	84.6±0.830	84.76±0.393	84.97±0.062
z_0	2±0.000	2±0.000	2±0.000
n	719.9±0.060	718.5±2.29	719.69±0.57
f	2.59±0.012	2.53±0.017	2.58±0.018
E	0.2376±0.02	0.2386±0.02	0.237±0.030
Q	0.0372±0.005	0.0356±0.0032	0.0369±0.0041
T	354.89±0.250	356.2±0.340	355.07±0.403

Firstly, the multi-objective intelligent algorithm is used to solve Eq. 9, and then multiple groups of non-dominated solutions are generated. Each method is run five times, and the average value and standard deviation of each result are observed. If users pay more attention to time-consuming, the optimal control parameters generated by each algorithm and the average value (Avg) and standard deviation (Std) of the results are shown in Table 1. A box diagram of the process comparison of the optimal parameters of the three algorithms is generated, as shown in Figure 3.

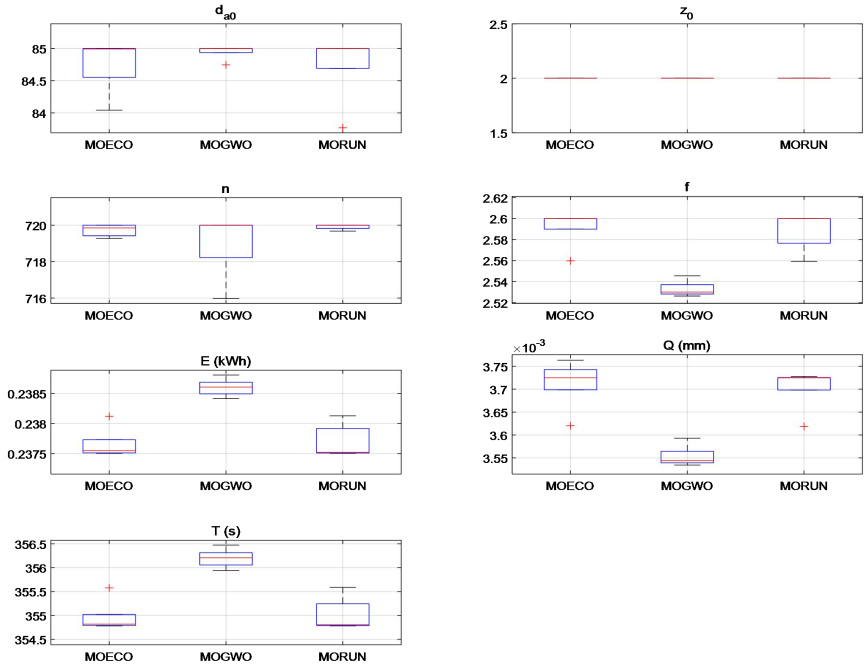

Figure 3. Box diagram of optimal control parameters for each method

From Figure 3 and Table 1, considering the data of Avg, the three algorithms yield consistent results in terms of z_0,

while differences exist in other indicators. With the data of MOECO as the reference value, in terms of d_{a0}, the relative deviations of MOGWO and MORUN are 0.19% and 0.44% respectively. In terms of n, the relative deviations of MOGWO and MORUN are 0.19% and 0.03% respectively. In terms of f, the relative deviations of MOGWO and MORUN are 2.32% and 0.39% respectively. In terms of E, the relative deviations of MOGWO and MORUN are 0.42% and 0.25% respectively. In terms of Q, the relative deviations of MOGWO and MORUN are 4.3% and 0.81% respectively. In terms of T, the relative deviations of MOGWO and MORUN are 0.37% and 0.05% respectively. From the perspective of Avg, the proposed algorithm performs more closely to MORUN, and ranks first in the target T, with very small gaps from the other two algorithms in E and Q.

Considering the data of Std, the three algorithms also show consistent results in z_0, and MOECO and MOGWO have the same value in E. In terms of d_{a0}, the relative deviations of MOGWO and MORUN are 52.65% and 92.53% respectively. In terms of n, the relative deviations of MOGWO and MORUN are 3716.67% and 850% respectively. In terms of f, the relative deviations of MOGWO and MORUN are 41.67% and 50% respectively. In terms of E, the relative deviations of MOGWO and MORUN are 0% and 0.5% respectively. In terms of Q, the relative deviations of MOGWO and MORUN are 36% and 18% respectively. In terms of T, the relative deviations of MOGWO and MORUN are 36% and 61.2% respectively. From the perspective of Std, there are significant gaps among the three algorithms. MOECO ranks first or ties for first in z_0, n, f, E, and T, which confirms the stability of the proposed method. However, as can be seen from Figure 3, there is an outlier in the proposed method, indicating some room for improvement.

In conclusion, MOECO can well solve engineering problems such as hobbing parameter calculation, and has significant advantages over MOGWO and MORUN.

5. Conclusions

To enable ECO to address multi-objective optimization problems, MOECO is proposed using Pareto improvement and roulette wheel selection strategy in this study. To verify the application efficacy of MOECO in hobbing control parameter optimization, MOECO was used to solve the control parameter optimization model, and the comparative experiments were conducted with MOGWO and MORUN. Through multiple algorithms runs, MOECO exhibits superior stability, and outperforms the comparative algorithms in optimizing energy consumption and processing time, illustrating the feasibility and effectiveness of MOECO in the control parameter optimization of hobbing.

Author Contributions: Conceptualization, W.C.; methodology, W.C. and G.W.; validation, T.P. and Y.J.; writing—original draft preparation, G.W.; writing—review and editing, W.C.; project administration, J.N.; funding acquisition, W.C.

Funding: This work research was funded by the National Natural Science Foundation of China, grant number 52305532 and Fundamental Research Funds for the Central Universities of China, grant number B230201023 and Jiangsu Key Laboratory of Power Transmission & Distribution Equipment Technology, grant number 2023JSSPD08 and Jiangsu Youth Science and Technology Talent Support Project, grant number JSTJ-2024-502.

Conflicts of Interest: The authors declare that the research was conducted without any commercial or financial relationships that could be construed as a potential conflict of interest.

References

[1] Hakim, W.L.; Fadhillah, M.F.; Won, J.-S.; Park, Y.-C.; Lee, C.-W. Advanced time-series InSAR analysis to estimate surface deformation and utilization of hybrid deep learning for susceptibility mapping in the Jakarta metropolitan region. *GISci. Remote Sens.* **2025**, *62*(1), 2465349.

[2] Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*(3), 46–61.

[3] Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872.

[4] Gharehchopogh, F.S.; Namazi, M.; Ebrahimi, L.; Abdollahzadeh, B. Advances in Sparrow Search Algorithm: A comprehensive survey. *Arch. Comput. Methods Eng.* **2022**, *30*(1), 427–455.

[5] Ma, H.; Zhang, Y.; Sun, S.; Liu, T.; Shan, Y. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artif. Intell. Rev.* **2023**, *56*(12), 15217–15270.

[6] Dhiman, G.; Kumar, V. Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems. *Knowl.-Based Syst.* **2018**, *150*, 175–197.

[7] El-rayes, K.; Kandil, A. Time-Cost-Quality Trade-Off

Analysis for Highway Construction. *J. Constr. Eng. Manag.* **2005**, *131*(4), 477–486.

[8] Ozcan-Deniz, G.; Zhu, Y.; Ceron, V. Time, Cost, and Environmental Impact Analysis on Construction Operation Optimization Using Genetic Algorithms. *J. Manag. Eng.* **2012**, *28*(3), 265–272.

[9] Parente, M.; Cortez, P.; Gomes Correia, A. An evolutionary multi-objective optimization system for earthworks. *Expert Syst. Appl.* **2015**, *42*(19), 6674–6685.

[10] Ozcan-Deniz, G.; Zhu, Y. Multi-objective optimization of greenhouse gas emissions in highway construction projects. *Sustain. Cities Soc.* **2017**, *28*, 162–171.

[11] Liu, L.; Zhang, C. Research on multi-objective optimization of construction engineering based on improved genetic algorithm. *Procedia Comput. Sci.* **2023**, *228*, 1086–1091.

[12] Zhang, H.; Xing, F. Fuzzy-multi-objective particle swarm optimization for time–cost–quality tradeoff in construction. *Autom. Constr.* **2010**, *19*(8), 1067–1075.

[13] Liu, S.; Tao, R.; Tam, C.M. Optimizing cost and CO2 emission for construction projects using particle swarm optimization. *Habitat Int.* **2013**, *37*, 155–162.

[14] Dhiman, G.; Kumar, V. Multi-objective spotted hyena optimizer: A Multi-objective optimization algorithm for engineering problems. *Knowl.-Based Syst.* **2018**, *150*, 175–197.

[15] Das, A.K.; Pratihar, D.K. A novel approach for neuro-fuzzy system-based multi-objective optimization to capture inherent fuzziness in engineering processes. *Knowl.-Based Syst.* **2019**, *175*, 1–11.

[16] Zouache, D.; Abdelaziz, F.B. Guided Manta Ray foraging optimization using epsilon dominance for multi-objective optimization in engineering design. *Expert Syst. Appl.* **2022**, *189*, 116126.

[17] Lian, J.; Zhu, T.; Ma, L.; Wu, X.; Heidari, A.A.; Chen, Y.; Chen, H.; Hui, G. The educational competition optimizer. *Int. J. Syst. Sci.* **2024**, *55*(15), 3185–3222.

[18] Cao, W.; Yu, Y.; Li, J.; Wu, D.; Ni, J.; Chen, X. High stability multi-objective decision-making approach of dry hobbing parameters. *J. Manuf. Process.* **2022**, *84*, 1184–1195.

[19] Cao, W.; Chen, X.; Ni, J. Fuzzy decision-making approach of hobbing tool and cutting parameters. *Eng. Appl. Artif. Intell.* **2023**, *125*, 106655.

[20] Cao, W.; Ni, J.; Jiang, B.; Ye, C. A three-stage parameter prediction approach for low-carbon gear hobbing. *J. Clean. Prod.* **2021**, *289*, 125777.

[21] Chen, X.; Li, X.; Li, Z. Control parameter optimization of dry hobbing under user evaluation. *J. Manuf. Process.* **2025**, *133*, 46–54.

[22] Ni, H.; Yan, C.; Ni, S.; Shu, H.; Zhang, Y. Multi-verse optimizer based parameters decision with considering tool life in dry hobbing process. *Adv. Manuf.* **2021**, *9*(2), 216–234.