

Article

Multi-Strategy Harris Hawks Optimization of Fuzzy Chance-Constrained Multi-Robot Hybrid Workshop Scheduling in Uncertain Environments

Mi Yang ^{1,*} , Zhan Zhang ¹, Xudong Zhu ²  and Jiguang Li ¹¹ School of Aerospace Engineering, Xihang University, Xi'an 710077, China² School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China

* Correspondence: yangmi@xaau.edu.cn

Abstract

Effective task allocation is fundamental to the success of heterogeneous multi-robot cooperative missions in smart manufacturing workshops, yet real-world operational uncertainties pose severe challenges to solution feasibility and mission robustness. Addressing these challenges, this paper focuses on the inspection and maintenance task allocation problem for heterogeneous mobile robot teams operating under fluctuating equipment maintenance time windows, variable task execution durations, and uncertain robot travel speeds caused by workshop congestion and payload variations. First, the aforementioned uncertain parameters are characterized using triangular fuzzy numbers, upon which a fuzzy chance-constrained programming model is constructed with the objective of minimizing total operational cost while ensuring constraint satisfaction under uncertainty. The proposed model simultaneously handles two types of critical constraints: the service time window constraint, which requires each task to be completed before its latest allowable service deadline, and the time sequence constraint, which enforces that each equipment inspection task must be completed prior to the corresponding maintenance task. Then, to tackle the inherent NP-hardness of this problem, a multi-strategy hybrid Harris Hawks Optimization algorithm incorporating differential evolution, termed MSHHODE, is proposed. In detail, three targeted enhancement mechanisms are introduced: a hunting enthusiasm factor that governs the dynamic balance between global exploration and local exploitation throughout the search process; an elite-assisted guidance strategy that stabilizes convergence by leveraging high-quality solutions to direct population evolution; and an adaptive differential evolution mechanism that reinforces global search diversity and mitigates premature convergence to local optima. Finally, simulation experiments conducted across multiple workshop-scale scenarios demonstrate that MSHHODE consistently outperforms benchmark algorithms across different key performance metrics under varied uncertain conditions, which validates the effectiveness and robustness of the proposed approach in solving complex, constrained allocation problems, offering a practical and reliable framework for real-world heterogeneous multi-robot task planning in smart manufacturing environments.



Academic Editor: Wen-Jer Chang

Received: 27 May 2026

Revised: 15 July 2026

Accepted: 23 July 2026

Published: 29 July 2026

Copyright: © 2026 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: multi-robot systems; task allocation; uncertain information; fuzzy chance-constrained; Harris Hawks Optimization; smart manufacturing

1. Introduction

In recent years, the rapid development of smart manufacturing technology has brought tremendous benefits to modern industrial production due to its advantages of high effi-

ciency, flexibility, and reduced labor dependency [1,2]. At the same time, the collaboration among multiple mobile robots can further enhance the capability of a single robot, enabling multi-robot systems to complete more complex tasks with higher efficiency, quality, and robustness [3–5]. It is noted that the core challenges of multi-robot cooperation, such as task allocation, path planning, and motion planning, play a decisive role in the execution of manufacturing missions and have received widespread attention [6–8].

Among them, task allocation refers to assigning a set of tasks to each robot with all constraints satisfied and optimizing a specific objective function to maximize overall operational benefits [9,10]. In smart manufacturing workshops, mobile robots are typically required to perform two sequential operations on target equipment: an inspection task (e.g., fault diagnosis, quality detection) followed by a maintenance task (e.g., repair, lubrication, component replacement), where the maintenance operation can only be initiated after the corresponding inspection is completed. However, when multiple heterogeneous robots perform tasks in actual workshop environments, various uncertain factors arise inevitably, including unknown fault complexity leading to uncertain task execution durations, dynamic production schedule adjustments causing fluctuating maintenance time windows, and workshop congestion or varying payloads resulting in uncertain robot travel speeds [11–13]. These uncertain factors impose more stringent requirements on the coordination of multi-robot systems [14]. However, most existing studies are based on static and deterministic task environments, often neglecting the instability of various factors in the workshop, such as the uncertainty of equipment maintenance duration and the impact of floor traffic conditions on robot traveling speed [15]. This leads to task allocation schemes optimized under predetermined deterministic conditions often failing to achieve the expected results in actual execution, or even causing equipment to exceed its maintenance deadline, resulting in unplanned production line interruptions [16].

Therefore, how to pre-consider the impact of uncertainties on the allocation results of heterogeneous multi-robot tasks, thereby improving the robustness of the allocation scheme, has become an unavoidable key issue in the practical deployment of multi-robot systems in smart manufacturing [17]. By predicting and handling various uncertainties during the pre-allocation phase, a stable task allocation scheme that is insensitive to uncertainty interference can be generated, which effectively reduces the impact of disturbances and minimizes the need for real-time rescheduling during task execution.

To address the aforementioned challenges, this paper investigates the heterogeneous multi-robot task allocation problem in smart manufacturing workshops under uncertain operational conditions, including stochastic task execution durations, fluctuating maintenance time windows, and uncertain robot travel speeds, subject to multiple complex mission constraints, including temporal sequencing and time window requirements. The main contributions of this work are summarized as follows.

(1) A fuzzy chance-constrained programming model is constructed based on fuzzy reliability theory, with the objective of minimizing total operational cost. By characterizing uncertain parameters through triangular fuzzy numbers and their associated membership functions, the proposed model circumvents the need for pre-estimation of probability distribution laws, which offers greater modeling flexibility and applicability under data-scarce conditions.

(2) To efficiently solve this NP-hard combinatorial optimization problem, a Multi-Strategy Hybrid Harris Hawks Optimization (HHO) algorithm integrated with Differential Evolution (MSHHODE) is proposed. Three synergistic enhancement mechanisms are embedded into the standard HHO framework: a hunting enthusiasm factor that adaptively regulates the transition between global exploration and local exploitation; an elite-assisted guidance strategy that leverages superior solutions to stabilize population convergence;

and an adaptive differential evolution mechanism that broadens search diversity and suppresses premature convergence to local optima.

(3) Comprehensive simulation experiments are conducted across multiple test scenarios, with systematic comparisons against representative benchmark algorithms. The results substantiate the effectiveness, robustness, and competitive superiority of the proposed MSHHODE in solving complex, constrained heterogeneous multi-robot task allocation problems under uncertainty.

The rest of this paper is organized as follows. Section 2 discusses the related work on task allocation for multiple robots under uncertainty. Section 3 formulates the considered multi-robot task allocation problem mathematically. The main structure of MSHHODE is illustrated in Section 4. Several simulations and comparisons are conducted in Section 5 to demonstrate the performance of MSHHODE. Section 6 concludes this paper.

2. Related Work

2.1. Smart Manufacturing Technology

With the development of technologies such as industrial Internet of Things, artificial intelligence, and digital twins, the manufacturing industry is shifting from a traditional scale efficiency orientation to a new paradigm of intelligence, flexibility, and service. Ref. [18] discussed the key technologies driving intelligent manufacturing and emphasized the progress of manufacturing technology that provides unprecedented design freedom. For modeling and simulation in a flexible manufacturing system (FMS), ref. [19] described the use of dedicated analysis and simulation software, through which the structure of the system can be established and specific technical and economic parameters can be determined for each processing and transport capacity. Different processing scenarios will be evaluated through virtual modeling and simulations in order to improve the performance and efficiency of the system. Ref. [20] demonstrated a practical implementation of a smart manufacturing application using a systematic approach based on a conceptual six-gear smart factory roadmap with connectivity, integration, and analytics stages to build a smart production management ecosystem using off-the-shelf technologies applied in precision manufacturing.

In addition, the AI approach to building flexible manufacturing systems has also been explosively researched [21,22]. In [23], the framework of the situation–actors–process (SAP) and learning–actions–performance (LAP) dimensions for the effective implementation of Gen AI is used to empower FMS. Ref. [24] provided a comprehensive overview of the current state of explainable AI, with a particular focus on recent advances and its critical role in enhancing trust and transparency in AI-driven smart manufacturing systems. Ref. [25] proposed the concept of human-centric smart manufacturing to entail a symbiotic human–machine relationship, and Trustworthy AI has been systematically reviewed as a key enabler through a progressive framework that addresses safety, transparency, and participatory updating across the product lifecycle.

Furthermore, the uncertainties within manufacturing systems are also gaining more and more attention. Ref. [26] combined generalized nets and intuitionistic fuzzy pairs, as measures of the uncertainty, for the modeling of flexible manufacturing systems and for the construction of simulation models. To address the uncertainties in the supply chain, ref. [27] proposed a hybrid fuzzy genetic algorithm framework that overcomes these constraints by combining the interpretability of fuzzy logic with the global search efficiency of genetic algorithms. Ref. [14] proposed a digital twin framework based on deep reinforcement learning for rapid response and system interaction to the uncertainty of corresponding demands and production dynamics. This framework provides fast and adaptive virtual

reality interaction and dynamic batch publishing strategies, while reducing production costs and improving the satisfaction rate of production dynamics.

Overall, current research on smart manufacturing has made positive progress in system modeling, AI empowerment, and uncertainty response, with increasingly diverse technological paths. However, the cross-scenario universality of each method and its effectiveness in handling uncertain parameters in each scenario still need further verification.

2.2. Handling Parameter Uncertainty

Uncertainty is a major characteristic of realistic smart manufacturing workshop environments, making the study of heterogeneous multi-robot task allocation under uncertain conditions of great significance for practical industrial deployment. In workshop settings, uncertainty arises from multiple sources, including unpredictable equipment fault complexity, dynamically adjusted production schedules, and fluctuating robot travel speeds caused by congestion and varying payload conditions. These uncertainties directly affect the feasibility and optimality of pre-planned task allocation schemes, potentially leading to missed maintenance deadlines and unplanned production disruptions. Currently, most multi-robot task allocation studies are based on deterministic environments and cannot adequately handle the impact of such uncertainties on allocation schemes. Typically, uncertain parameters are incorporated into the optimization model as random variables or imprecise quantities. Depending on how these uncertain parameters are described and processed, there are three main approaches to addressing the impact of uncertain factors on task allocation: stochastic programming, robust optimization, and fuzzy programming.

Stochastic programming, as a commonly used method for dealing with uncertainty, introduces uncertain random variables described by probability distribution functions, and solves them by establishing a random-variable model of the optimization objective function [28]. Ref. [29] proposed a two-stage stochastic 0-1 integer linear programming model to equitably optimize ambulance allocation in the Basque Country, balancing multi-interval response times between urban and rural areas through stochastic programming. Ref. [30] developed a two-stage stochastic optimization model that jointly allocates Emergency Mobility Facilities (EMFs) and designed road networks for disaster response, addressing uncertainties in demand and congestion through reformulation into a tractable mixed-integer linear program validated with real-world data. However, the application of stochastic programming methods requires estimating the distribution of uncertain parameters, and the estimated distribution needs to be close enough to the actual distribution. In actual environments, though, it is difficult to obtain the probability of uncertain events [31].

Therefore, some scholars have used robust optimization methods to solve the optimization problem under uncertain conditions. By controlling the conservative parameters of decision-making, the uncertain optimization model is transformed into a deterministic linear programming model [32]. Since there is no need to assume the distribution of uncertain parameters, it is easier to solve than the stochastic programming model. Ref. [33] proposed a multi-objective mathematical model based on multi-scenario robust optimization for the optimal design and facility location allocation of blood supply chain networks after natural disasters, aiming to minimize shortages and costs. Ref. [34] proposed a two-stage robust optimization model integrating prospect theory for the berth allocation and quay crane assignment problem under uncertainty, along with an accelerated algorithm for efficient large-scale solutions. Ref. [35] proposed a scenario-based robust optimization model for sustainable distributed flow-shop scheduling under uncertainty, solved by an Adaptive Large Neighborhood Search algorithm to enhance energy efficiency, job creation, and workday utilization. However, the task allocation schemes solved by robust optimization

methods are often overly conservative, making it difficult to balance system performance and protection against uncertainty.

In addition to the two methods mentioned above, the fuzzy programming method describes the uncertain parameters in the form of fuzzy sets also widely applied to solve task allocation problems under uncertain conditions. It only needs to select the membership function of the uncertain variables based on prior knowledge and emphasizes the possibility of satisfying the constraints [36]. By transforming the objective function and constraints of the fuzzy programming into a general form of a programming model, the fuzzy optimal solution can be obtained. Ref. [37] proposed a Stackelberg game-based fuzzy optimization model for tugboat scheduling under uncertainty, solved by an enhanced seagull algorithm and validated with real port data. Ref. [38] developed a fuzzy multi-constraint programming model and a dynamic strategy with an enhanced grey wolf optimization algorithm for heterogeneous multi-robot dynamic task assignment under uncertainty, demonstrating 5–30% performance improvement. The fuzzy programming technique with a linear membership function is applied in [39] to generate efficient and non-dominated allocations with better optimal values and degrees of satisfaction of objective functions for different values of parameters α and β for fuzzy preferences. Ref. [40] developed a fuzzy multi-criteria decision-making model for resilient supplier selection under dynamic conditions, optimized by an enhanced wolf pack algorithm for order allocation, and validated its feasibility through a case study in the new energy vehicle industry.

The above three types of methods focus on how to make optimal decisions under uncertainty, but in addition to making reasonable decision plans, it is also necessary to quantitatively evaluate the reliability of existing plans. Fuzzy reliability theory, on the other hand, focuses on how reliable the design scheme is, which means the ability of the system to complete the specified functions under specified conditions and within a specified time [41]. This theory introduces fuzzy sets into traditional reliability engineering to address the problem of difficulty in obtaining accurate failure probabilities.

At the methodological level, fuzzy reliability is mainly achieved through fuzzy fault tree analysis. Firstly, for example, the failure probability of basic events is represented by fuzzy numbers, such as triangular fuzzy numbers, and then the fuzzy failure probability and fuzzy reliability of the top event are calculated. Ref. [42] investigated the mixed statistical fuzzy reliability evaluation composed of intuitionistic fuzzy sets (IFSs) and interval-valued intuitionistic fuzzy sets. In [43], a fault tree analysis based on the fuzzy set theory is presented and applied to a wind energy system. The fault probabilities are regarded as a fuzzy number, the fuzzy failure rate is known by using fuzzy rules, and the risk analysis method is applied based on the fuzzy risk index (FRI) to know the exact impact of every basic event on the top event.

In addition, the stress strength fuzzy interference model and the combination method of fuzzy fault tree with Bayesian network and Petri net have also been applied in the reliability evaluation of complex manufacturing systems. For example, ref. [44] proposed a reliability analysis method based on fuzzy polymorphic Bayesian network (FPBN), which utilizes a multi-state fault tree to construct a polymorphic Bayesian network (BN), combining traditional BN techniques with a consideration of multiple fault states and fuzzy failure rates to handle inaccurate fault data and unclear logical relationships. Ref. [45] combined fuzzy BN with result simulation, using leakage noise or gate, intuitionistic fuzzy set theory, and similarity aggregation for expert inspiration to fill the conditional probability table, and used Safeti software for result analysis.

2.3. Recent Developments of Harris Hawks Optimization

Since its proposal by Heidari et al. in 2019, the HHO algorithm has rapidly become a research hotspot in the field of swarm intelligence optimization due to its unique mechanism of simulating Harris hawk cooperative hunting behavior and the advantage of not requiring gradient information [46,47]. However, standard HHO still faces inherent limitations such as slow convergence speed, insufficient population diversity, and susceptibility to falling into local optima in high-dimensional complex problems.

To overcome the above shortcomings, researchers have continuously improved HHO from multiple dimensions such as parameter control, strategy improvement, and hybrid enhancement, and have derived a large number of variant algorithms. For example, ref. [48] integrated an improved HHO (IIHHO) algorithm, which introduces an intermittent energy regulator, a regularized vector mechanism, a refined Lévy flight search scope, and a Cardano formula-based step-size adjustment to collectively enhance local search capability, escape from local optima, and increase solution accuracy. Ref. [49] improved Harris Hawks optimization (IHHO) by modifying the exploration phase with random location-based habitats and refining exploitation via three strategies, while employing logarithm- and exponential-based random value generation to escape local optima, demonstrating superior performance across diverse benchmarks and engineering problems. The proposed Chaotic Harris Hawks Optimization (CHHO) in [50] enhances the standard HHO by introducing chaotic maps at the initialization phase to improve population diversity and incorporating simulated annealing on the current best solution to strengthen exploitation, thereby addressing the local optima and diversity limitations for feature selection problems. Ref. [51] applied chaotic maps at the initialization phase of HHO to enhance the population diversity in the search space, and used the simulated annealing (SA) algorithm to the current best solution to improve HHO exploitation. Ref. [52] proposed a Modified HHO (MHHO) that integrates Leader-Guided Perching (LGP) to accelerate movement toward promising regions, an Adaptive Deception Factor (ADF) to dynamically scale Lévy-flight intensity to escape deceptive local optima, and a Hierarchical Attack Strategy (HAS) to replaces the hard besiege phase with a mentor-guided exploitation step to mitigate swarm clumping and refine local search. The comparison table of the HHO algorithm literature is shown in Table 1.

Table 1. Feature comparison of representative HHO variants and the proposed MSHHODE.

Reference	Improvement Mechanism	Stage Coverage			Adaptive Switching	Population-Level Cooperation
		Exploration	Transition	Exploitation		
[47]	Energy E -driven four besieging strategies	✓	×	✓	×	×
[48]	Intermittent energy regulator + regularized vector + refined Lévy step	×	×	✓	×	×
[49]	Random habitat-based exploration + log/exponential perturbation	✓	×	×	×	×
[50]	Chaotic map initialization + simulated annealing on best solution	✓	×	✓	×	× [†]
[51]	Chaotic map initialization + simulated annealing	✓	×	✓	×	× [†]
[52]	Leader-Guided Perching + Adaptive Deception Factor + Hierarchical Attack Strategy	×	×	✓	×	×
Proposed	Hunger-energy joint hunting enthusiasm factor + elite-group centroid guidance + multi-factor adaptive DE-triggering	✓	✓	✓	✓	✓

The ✓ indicates that the algorithm possesses or explicitly incorporates the corresponding improvement characteristic, while × indicates that it does not. [†] Simulated annealing is applied only to the single best individual, not at the population level.

As shown in Table 1, existing HHO variants typically improve only a single stage of the search process and retain the fixed energy-threshold switching criterion, with any guidance mechanisms applied to a single individual rather than the population. In contrast, MSHHODE jointly covers all three stages through the hunting enthusiasm factor, adopts a dynamically adaptive switching criterion, and achieves population-level cooperation via the elite-group guidance and adaptive DE-triggering mechanisms, offering a more robust basis for the NP-hard task allocation problem studied in this paper.

3. Problem Formulation

In this section, the task allocation problem of heterogeneous multi-robot systems under uncertain information is discussed, where the problem background, the models of robots, the tasks, and the time with complex constraints are first specified. On this basis, the uncertain optimization problem is further transformed into a fuzzy chance-constrained programming model using fuzzy uncertainty theory.

3.1. Scenario Description

Consider a large-scale smart manufacturing workshop where M heterogeneous mobile robots (inspection robots, maintenance robots, and combined inspection–maintenance robots) collaboratively perform inspection and maintenance tasks on N target workstations or production equipment units on the shop floor. Prior to task execution, the workshop management system has obtained relevant information about the target equipment through IoT sensors and manufacturing execution systems (MESs), including equipment type, maintenance priority, and location. However, due to measurement noise in sensor readings, dynamic adjustments in production schedules, and varying robot performance caused by workshop congestion and payload fluctuations, the known task execution duration, the latest allowable maintenance time of the equipment, and the travel speed of the robots are subject to uncertainty, which is described by triangular fuzzy numbers.

It is noted that the final task allocation scheme must satisfy both the temporal sequencing constraint and service time window constraint to reduce the sensitivity of the robot system to uncertainties during task execution. Specifically, the temporal sequencing constraint requires that each equipment inspection task must be completed before the corresponding maintenance task can be initiated, since maintenance operations depend on the diagnostic results obtained during inspection. In addition, the service time window constraint requires that each task must be completed by the assigned robot before the latest allowable service time s of the target equipment, where the latest allowable service time can be determined by the equipment's criticality level or specified by the production manager based on the overall manufacturing progress.

For ease of analysis, the following assumptions are made:

- There are no restricted areas or sudden dynamic obstacles on the shop floor that would permanently block robot navigation.
- All robots operate on the same floor level and share the same global workshop map and equipment status information.
- Each robot can perform multiple tasks sequentially, but can only execute one task at any given time.
- Inspection robots do not consume any consumable materials during inspection tasks; both combined robots and maintenance robots carry sufficient materials and tools to complete maintenance tasks for all target equipment. Therefore, this paper does not consider the impact of material resource constraints on task allocation schemes.
- All target equipment units are within the operational range of the deployed robots.

(1) Heterogeneous robot model

There are M mobile robots of three types (inspection robots, maintenance robots, and combined inspection–maintenance robots), $\mathbf{U} = \{U_1, U_2, \dots, U_M\}$. Inspection robots can only perform inspection tasks, maintenance robots can only perform maintenance tasks, and combined inspection–maintenance robots can perform both types of tasks. Robot attributes are represented by the tuple $\langle Uid_i, Utype_i, Uval_i, AT_i, Upos_i, v_i \rangle$, including the robot ID, type, intrinsic operational value, probability of task failure due to mechanical malfunction or navigation error, location, and travel speed. The travel speed of the robot is represented by a triangular fuzzy number $\tilde{v}_i = (v_{1i}, v_{2i}, v_{3i})$, where v_{2i} denotes the most likely speed, and v_{1i} and v_{3i} denote the lower and upper bounds of the speed range, respectively.

(2) Equipment task model

Suppose there are N target equipment units in a two-dimensional smart manufacturing workshop, denoted by $\mathbf{T} = \{T_1, T_2, \dots, T_N\}$. Due to the two-stage nature of equipment servicing, where each equipment unit requires both an inspection task and a maintenance task, there are $2N$ tasks available in total.

$$\sum_{j=1}^N \sum_{h=1}^2 |T_{jh}| = 2N, \quad (1)$$

where T_{jh} represents the h -th type of task ($h = 1, 2$) for target equipment T_j . $h = 1$ represents a inspection task, and $h = 2$ represents a maintenance task. It should be noted that according to the operational requirements of equipment servicing, a maintenance task can only be initiated after the corresponding inspection task has been completed, since the maintenance procedure depends on the diagnostic results obtained during inspection.

The tuple $\langle Tid_j, Tval_j, PK_j, Tpos_j, s_j, Ct_j^D, Ct_j^A \rangle$ is used to represent the equipment ID, maintenance priority value, probability of robot task failure at equipment, coordinates of position, latest allowable service time, and durations of inspection and maintenance tasks for equipment T_j . This paper assumes that maintenance benefits are only realized after both the inspection and maintenance tasks of a given equipment unit have been successfully completed. The execution duration $\tilde{Ct}^D$ of inspection task T_{j1} for equipment T_j , the execution duration $\tilde{Ct}^A$ of maintenance task T_{j2} for target T_j , and the latest allowable service time s are all subject to uncertainty and are expressed using triangular fuzzy numbers as follows:

$$\begin{aligned} \tilde{Ct}^D &= (Ct_1^D, Ct_2^D, Ct_3^D) \\ \tilde{Ct}^A &= (Ct_1^A, Ct_2^A, Ct_3^A) \end{aligned} \quad (2)$$

$$\tilde{s} = (s_1, s_2, s_3). \quad (3)$$

3.2. Processing of Uncertain Information

Compared to a deterministic task environment, the uncertain task execution duration, the latest allowable service time of the target equipment, and the travel speed of the robots make the time window and temporal constraints in the task allocation problem more difficult to satisfy. Specifically, the accumulation of uncertainties along the task execution chain increases the probability that a robot will fail to complete its assigned tasks within the specified service time window, potentially leading to missed maintenance deadlines and consequent production disruptions. Therefore, this paper constructs a fuzzy chance-constrained programming model based on fuzzy reliability theory, incorporating the aforementioned uncertain parameters into the temporal sequencing and service time window constraints of the optimization problem for systematic processing.

In this paper, triangular fuzzy numbers are adopted here for three main reasons. Firstly, unlike interval numbers, which consider all values within the boundary to be as equal as possible, triangular fuzzy numbers capture the most likely values and pessimistic/optimistic boundaries, which matches the fact that the nominal task duration, time window, and rated robot speed are more likely than their extremum. Moreover, compared to intuitionistic fuzzy numbers, trapezoidal fuzzy numbers are more suitable for parameters with a flat center value range or explicit expert hesitation. In addition, triangular fuzzy numbers do not require accurate probability distribution estimation from historical data; only three representative values need to be obtained from equipment specifications or expert experience.

3.2.1. Fuzzy Reliability Theory

Fuzzy reliability theory extends classical reliability methods to handle systems with imprecise or uncertain data, using fuzzy sets to model parameters like failure rates when historical data is limited or subjective. This approach supports reliability analysis under vague conditions, such as human-influenced operations or emerging technologies, enabling more flexible risk assessment and decision-making in complex engineering and safety-critical environments.

For the triangular fuzzy numbers $\tilde{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$ and $\tilde{\theta} = (\theta_1, \theta_2, \theta_3)$, their basic operation rules are shown in Equation (4), and their credibility calculation in credibility theory is shown in Equation (5):

$$\begin{cases} \tilde{\lambda} + \tilde{\theta} = (\lambda_1 + \theta_1, \lambda_2 + \theta_2, \lambda_3 + \theta_3) \\ K\tilde{\lambda} = (K\lambda_1, K\lambda_2, K\lambda_3) \\ \tilde{\lambda}^{-1} = (1/\lambda_3, 1/\lambda_2, 1/\lambda_1) \end{cases} \quad (4)$$

$$\begin{aligned} c_r(\tilde{\lambda} \leq \tilde{\theta}) &= c_r\{(\lambda_1 - \theta_3, \lambda_2 - \theta_2, \lambda_3 - \theta_1) \leq 0\} \\ &= \begin{cases} 0, & \lambda_1 \geq \theta_3 \\ \frac{\theta_3 - \lambda_1}{2(\theta_3 - \lambda_1 + \lambda_2 - \theta_2)}, & \lambda_1 \leq \theta_3, \lambda_2 \geq \theta_2 \\ \frac{\lambda_3 - \theta_1 - 2(\lambda_2 - \theta_2)}{2(\theta_2 - \lambda_2 + \lambda_3 - \theta_1)}, & \lambda_2 \leq \theta_2, \lambda_3 \geq \theta_1 \\ 1, & \lambda_3 \leq \theta_1 \end{cases} \end{aligned} \quad (5)$$

where $c_r \in [0, 1]$, and a larger c_r indicates a greater likelihood of $\tilde{\lambda} \leq \tilde{\theta}$. By introducing a preference value $c_r^* \in [0, 1]$ to characterize the decision-maker's attitude, we use $c_r \geq c_r^*$ to constrain the credibility of $\tilde{\lambda} \leq \tilde{\theta}$. The smaller the value of c_r^* , the more likely the decision-maker is to risk failure by making a decision based on $\tilde{\lambda} \leq \tilde{\theta}$; conversely, the larger the c_r^* , the more the decision-maker would rather reconsider the decision than make a decision based on $\tilde{\lambda} \leq \tilde{\theta}$.

3.2.2. Service Time Window Constraints

Because the smart manufacturing workshop operates under dynamic production schedules and strict equipment maintenance requirements, it is necessary to complete the inspection and maintenance tasks for each target equipment unit before a specific deadline, i.e., the equipment units are subject to service time window constraints. The time window here can be determined by the criticality level of the equipment to the production line, or specified by the production manager based on the overall manufacturing progress and maintenance plan. Suppose that a certain robot i completes the first $k-1$ tasks in its sequence $\mathbf{a}_i$ of tasks to be executed, then the time for robot i to reach the k -th task position is shown in Equation (6).

$$\begin{aligned} \tilde{B}T_{ik} &= \sum_{l=1}^{k-1} Ct_l + \sum_{l=0}^{k-1} \frac{D_{l,l+1}}{\tilde{v}_i} \\ &= \begin{cases} \sum_{l=1}^{k-1} Ct_{l1} + \sum_{l=0}^{k-1} \frac{D_{l,l+1}}{\tilde{v}_{i3}}, & l = 1, \dots, k \\ \sum_{l=1}^{k-1} Ct_{l2} + \sum_{l=0}^{k-1} \frac{D_{l,l+1}}{\tilde{v}_{i2}}, & l = 1, \dots, k, \quad l = 1, \dots, k \\ \sum_{l=1}^{k-1} Ct_{l3} + \sum_{l=0}^{k-1} \frac{D_{l,l+1}}{\tilde{v}_{i1}}, & l = 1, \dots, k \end{cases}, \end{aligned} \quad (6)$$

where $Ct_l = (Ct_{l1}, Ct_{l2}, Ct_{l3})$ is the execution time of the l -th task in the task list, determined according to the corresponding target equipment and its task type. Let $D_{i,j}$ be the distance between tasks i and j , expressed in Euclidean distance through Equation (7), that is, the time to reach the k -th task is the sum of the execution times of the first $k-1$ tasks and the travel time of the robot from the starting point to the positions of the first k tasks in sequence.

$$D_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \quad (7)$$

When robot i executes the k -th task in its task list, based on the reliability measure theory, the reliability $c_r\{\tilde{B}T_{ik} \leq s_k\}$ of its arrival time at task k being less than the latest execution time s_k of task k is denoted as c_{r1} , and $c_{r1} \in [0, 1]$. The larger c_{r1} is, the greater the probability that robot i 's arrival time at task k is earlier than the latest execution time of task k , i.e., the greater the probability of satisfying the service time window constraint of task k .

Introducing the decision-maker's preference value $c_{r1}^* \in [0, 1]$, if $c_{r1} \geq c_{r1}^*$, then robot i continues to execute the next task k in its task list, i.e., the reliability of satisfying the target equipment service time window constraint is greater than the preference value. If $c_{r1} < c_{r1}^*$, then the reliability of robot i satisfying the target equipment service time window constraint is less than the preference value, and robot i abandons executing the next task k in its task list.

3.2.3. Temporal Sequencing Constraint

For each target equipment, inspection and maintenance tasks must be completed sequentially, i.e., the time sequence constraint must be met. For target equipment j , let $T_{j1} = a_{i_1 L_1}$, $T_{j2} = a_{i_2 L_2}$, that is, its inspection task T_{j1} is assigned to the L_1 position of the robot i_1 task list and the maintenance task T_{j2} is executed by the L_2 position of the robot i_2 task list. Then, the task completion time of T_{j1} can be obtained in Equation (8):

$$ET_{i_1 L_1} = \tilde{B}T_{i_1 L_1} + \tilde{C}t_{L_1} = \sum_{l=1}^{L_1} Ct_l + \sum_{l=0}^{L_1-1} \frac{D_{l,l+1}}{\tilde{v}_i}. \quad (8)$$

Based on the credibility measure theory, the credibility $c_\gamma\{ET_{i_1 L_1} \leq BT_{i_2 L_2}\}$ of the completion time of inspection task T_{j1} on target equipment j being less than the start time of maintenance task T_{j2} is denoted as $c_{\gamma 2}$, and $c_{\gamma 2} \in [0, 1]$. The larger $c_{\gamma 2}$ is, the greater the probability that the completion time of the inspection task of target equipment j is less than the start time of the maintenance task, that is, the greater the probability that the temporal constraints of target equipment j are satisfied.

Introducing a preference value $c_{\gamma 2}^* \in [0, 1]$, if $c_{\gamma 2} \geq c_{\gamma 2}^*$, then the probability that the completion time of the inspection task of target equipment j is lower than the start time of the maintenance task is greater than the preset preference value, that is, the credibility of satisfying the target's temporal constraints is greater than the preference value. If $c_{\gamma 2} < c_{\gamma 2}^*$, then the credibility of satisfying the temporal constraints of target equipment j is less than the preference value.

3.3. Cost and Benefit Function

In actual smart manufacturing workshop operations, robots navigating through densely equipped shop floors are subject to the risk of task failure due to mechanical malfunctions, navigation errors, or collisions with workshop obstacles and other robots. Furthermore, robots consume energy during task execution, and excessive travel distances lead to increased operational costs. Therefore, this paper considers the costs of heterogeneous multi-robot execution of inspection and maintenance tasks in two aspects: (1) the operational risk cost C_1 incurred during robot task execution, reflecting the potential loss caused by robot malfunction or task failure; and (2) the travel cost C_2 incurred during robot navigation between task locations, reflecting the energy consumption and wear associated with robot movement.

Let PS_i be the probability that robot i successfully completes its task at equipment j without malfunction or navigation failure, then the probability of task failure occurring at equipment j is $PT_j = 1 - PS_j$, which reflects the operational risk imposed by the complexity of the equipment and the difficulty of access in the surrounding workshop environment. Therefore, the operational risk cost generated by a single robot i performing m tasks is as shown in Equation (9):

$$C_1 = Uval_i * \left(1 - \prod_{j=1}^m (1 - PK_j) \right). \quad (9)$$

Assuming that robots tend to be assigned to tasks closer to their current location in order to minimize energy consumption and reduce wear on mechanical components, the travel cost can be expressed in Equation (10):

$$C_2 = \frac{D_{i,j}}{D_{\max,j}}, \quad D_{\max,j} = \max_{i \in U} (D_{i,j}), \quad (10)$$

where $D_{\max,j}$ represents the maximum travel distance among all robots to reach task j , serving as a normalization reference for evaluating the relative travel burden of each robot assignment. The maintenance benefit of a heterogeneous multi-robot system completing a servicing task refers to the operational value recovered by successfully executing both the inspection and maintenance tasks for a target equipment unit, determined by the equipment's maintenance priority value and the robot's task success probability. Its magnitude reflects the criticality of the equipment to the production line and the execution capability of the assigned robot, guiding the final allocation result towards maximizing overall production efficiency and equipment health during decision optimization. Let At_i be the probability of the i -th robot successfully completing task j , then the maintenance benefit obtained from servicing the target equipment is shown in Equation (11):

$$C_3 = Val_j * At_i. \quad (11)$$

3.4. Problem Mathematical Modeling

The task allocation problem of heterogeneous multi-robot systems under conditions with uncertain task execution duration, latest allowable service time of target equipment, and robot travel speed is modeled as a fuzzy chance-constrained model. First, the problem is transformed into a single-objective fuzzy chance-constrained model using a linear weighting method, and then the dimensions are converted to values within the set $[0, 1]$ using a linear scaling method. The modeled objective function of the fuzzy chance-constrained programming is Equation (12):

$$\min f = \sum_{i=1}^M \sum_{j=1}^N \sum_{h=1}^2 \left(\omega_1 \frac{C_1}{\max_{i \in U} Uval_i} + \omega_2 C_2 - \omega_3 \frac{C_3}{\max_{j \in T} Tval_j} \right) x_{ijh}, \quad (12)$$

s.t.

$$c_r \{BT_{ik} \leq s_k\} \geq c_{r1}^*, \quad \forall i \in \mathbf{U}, \forall k \in 1, \dots, |a_i|, \quad (13)$$

$$c_r \{BT_{jh} + Ct_{jh} \leq ET_{j(h+1)}\} \geq c_{r2}^*, \quad h = 1, \forall j \in \mathbf{T}, \quad (14)$$

$$\sum_{i=1}^M \sum_{j=1}^N \sum_{h=1}^2 x_{ijh} = 2N, \quad \forall i \in \mathbf{U}, \forall j \in \mathbf{T}, \quad (15)$$

$$\sum_{i=1}^M x_{ijh} = 1, \quad \forall j \in \mathbf{T}, h \in \{1, 2\}, \quad (16)$$

$$V_i \leq V_{\max i}, \quad \forall i \in \mathbf{U}, \quad (17)$$

$$x_{ijh} \in \{0, 1\}, \quad \forall i \in \mathbf{U}, \forall j \in \mathbf{T}, h \in \{1, 2\}, \quad (18)$$

where different weight vectors are set to balance the influence of various factors on the allocation result in Equation (12). The weights ω_1 , ω_2 , and ω_3 represent the weights of the operational risk cost C_1 incurred during robot task execution, the travel cost C_2 incurred during robot navigation, and the maintenance benefit C_3 obtained from successfully completing the inspection and maintenance tasks, respectively. Different weight values reflect the different decision-making preferences of the production manager. Equation (13) indicates that the reliability of the allocation scheme satisfying the service time window constraint is not less than the subjective preference value c_{r1}^* ; Equation (14) indicates that the reliability of the allocation scheme satisfying the temporal sequencing constraint is not less than the subjective preference value c_{r2}^* , which is set to $c_{r1}^* = c_{r2}^* = c_r$ in this paper. Equation (15) ensures that both the inspection and maintenance tasks of each target equipment unit can be executed; Equation (16) means that each task can only be executed by one robot exactly once; Equation (17) indicates the operational travel constraint of each robot, where $V_{\max i}$ is the maximum operational travel of robot i ; and Equation (18) indicates that the decision variable x_{ijk} takes the value of 0 or 1.

The aforementioned model has a non-linear objective function and constraints, and the decision variables are non-negative integers, thus constituting a non-linear integer programming model. Since this model belongs to the NP-hard problem, the solution space grows exponentially with the problem size and the number of constraints. When using traditional exact algorithms to solve such problems, such as branch and bound and dynamic programming, the optimal solution cannot be obtained within a given time. Therefore, this paper proposes the MSHHODE algorithm to find the suboptimal solution to the model. Key parameters used in the fuzzy chance-constrained model are displayed in Table 2.

Table 2. Key parameters used in the fuzzy chance-constrained model.

Symbol	Description
M	Number of heterogeneous mobile robots
N	Number of target equipment units
U_i	The i -th mobile robot
T_j	The j -th target equipment unit
T_{jh}	The h -th type of task ($h = 1$: inspection; $h = 2$: maintenance) for equipment T_j
$U_{id_i}, U_{type_i}, U_{val_i}$	Robot ID, type, and intrinsic operational value
AT_i	Probability of task failure due to mechanical malfunction or navigation error of robot i
U_{pos_i}	Position of robot i
$\tilde{v}_i = (v_i^1, v_i^2, v_i^3)$	Triangular fuzzy travel speed of robot i
T_{id_j}, T_{val_j}	Equipment ID and maintenance priority value
PK_j	Probability of robot task failure at equipment T_j
T_{pos_j}	Position of equipment T_j
$\tilde{s} = (s_1, s_2, s_3)$	Triangular fuzzy latest allowable service time
$\tilde{Ct}^D = (Ct_1^D, Ct_2^D, Ct_3^D)$	Triangular fuzzy execution duration of inspection task
$\tilde{Ct}^A = (Ct_1^A, Ct_2^A, Ct_3^A)$	Triangular fuzzy execution duration of maintenance task
$\tilde{\lambda} = (\lambda_1, \lambda_2, \lambda_3), \tilde{\theta} = (\theta_1, \theta_2, \theta_3)$	Generic triangular fuzzy numbers
$cr\{\cdot\}$	Credibility measure
c_r^*	Decision-maker's preference value for credibility constraints
$\tilde{BT}_{ik}$	Fuzzy arrival time of robot i at the k -th task in its sequence
$D_{i,j}$	Euclidean distance between tasks i and j
$\tilde{ET}_{i,l1}$	Fuzzy completion time of task at position $l1$ of robot i
C_1	Operational risk cost
C_2	Travel cost
C_3	Maintenance benefit
PS_i	Probability that robot i successfully completes a task without malfunction or navigation failure
PT_i	Probability of task failure at equipment j ($PT_i = 1 - PS_i$)
$D_{max,j}$	Maximum travel distance among all robots to reach task j
At_i	Probability of robot i successfully completing task j
$\omega_1, \omega_2, \omega_3$	Weights of operational risk cost, travel cost, and maintenance benefit in the objective function
x_{ijh}	Binary decision variable indicating whether task h of equipment j is assigned to robot i
V_{max_i}	Maximum operational travel of robot i

4. Proposed Method

4.1. Harris Hawks Optimization (HHO) Algorithm

In nature, Harris hawks use their sharp eyes to scout the environment and efficiently perform cooperative foraging during tracking, encirclement, and attack phases. Heidari et al. [47] conducted in-depth research on the hunting behavior of Harris hawks and proposed the Harris Hawks Optimization algorithm (HHO), which includes three phases: global exploration phase, global exploration to local development phase, and local development phase. The main pseudocode of HHO is displayed in Algorithm 1. Experiments show that HHO performs well in unimodal, multimodal, and combinatorial optimization problems, especially demonstrating good scalability and robustness in high-dimensional problems and engineering optimization design.

Algorithm 1 Pseudo-code of HHO algorithm

Input: Unallocated tasks set $\mathcal{T}^U$
Output: Allocated tasks sequence $\mathcal{A}_i, i \in \mathcal{U}$

```

1: Initialize the random population  $X_i (i = 1, 2, \dots, N)$ 
2: while (stopping condition is not met) do
3:   Calculate the fitness values of hawks
4:   Set  $X_{\text{rabbit}}$  as the location of rabbit (best location)
5:   for each hawk ( $X_i$ ) do
6:     Update the initial energy  $E_0$  and jump strength  $J$ 
7:     Update the  $E$  by  $E = 2E_0(1 - \frac{t}{T})$ 
8:     if  $|E| \geq 1$  then
9:       Switch to exploration phase
10:      Update the location vector by exploration phase
11:    end if
12:    if  $|E| < 1$  then
13:      Switch to exploitation phase
14:      if  $r \geq 0.5$  and  $|E| \geq 0.5$  then
15:        Update the location vector by soft besiege
16:      else if  $r \geq 0.5$  and  $|E| < 0.5$  then
17:        Update the location vector by hard besiege
18:      else if  $r < 0.5$  and  $|E| \geq 0.5$  then
19:        Update the location vector by soft besiege with progressive rapid dives
20:      else if  $r < 0.5$  and  $|E| < 0.5$  then
21:        Update the location vector by hard besiege with progressive rapid dives
22:      end if
23:    end if
24:  end for
25: end while
26: return  $X_{\text{rabbit}}$ 

```

4.1.1. Exploration Phase

Assuming the population size is $PopSize$ and the dimension is Dim , define the position vector of the t -th generation population as $X_1(t), X_2(t), \dots, X_N(t)$, where the i -th individual is expressed as follows:

$$X_i(t) = (X_{i1}(t), X_{i2}(t), \dots, X_{iDim}(t)), \quad i = 1, \dots, PopSize. \quad (19)$$

During the exploration phase, the Harris hawk employs two strategies to search for prey, with both strategies having equal probability. The two search strategies are as follows:

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1|X_{\text{rand}}(t) - 2r_2X(t)|, & q \geq 0.5 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)), & q < 0.5 \end{cases} \quad (20)$$

where $X(t+1)$ is the position vector of the hawks in the next iteration t , $X_{\text{rabbit}}(t)$ is the position of the rabbit, $X(t)$ is the current position vector of the hawks, and r_1, r_2, r_3, r_4 , and q are random numbers in $[0, 1]$, which are updated in each iteration. LB and UB are the upper and lower bounds of the variables, respectively, $X_{\text{rand}}(t)$ is a randomly selected hawk from the current population, and X_m is the average position of the current population of hawks. The average position of the hawks is obtained using Equation (21):

$$X_m(t) = \frac{1}{N} \sum_{i=1}^N X_i(t). \quad (21)$$

4.1.2. Transition from Exploration to Exploitation

To model this step, the energy of a rabbit is modeled as

$$E = 2E_0 \left(1 - \frac{t}{MaxIter} \right), \quad (22)$$

where E indicates the escaping energy of the prey, $MaxIter$ is the maximum number of iterations, and E_0 is the initial state of its energy. When $|E| \geq 1$, the exploration phase begins; when $|E| < 1$, the exploitation phase begins.

4.1.3. Exploitation Phase

The algorithm's development phase includes four different search strategies to mimic the hunting process of a flock of hawks: soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives. The corresponding population update is performed based on the escape energy and the value of the random number r in the interval $(0, 1)$.

① Soft besiege

When $|E| \geq 0.5$ and $r \geq 0.5$, the prey still has enough energy to escape, and the flock of hawks will gently circle and chase the prey, exhausting it. The strategy expression is

$$X(t+1) = \Delta X(t) - E|JX_{rabbit}(t) - X(t)| \quad (23)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t), \quad (24)$$

where $\Delta X(t)$ is the difference between the position vector of the rabbit and the current location in iteration t , r_5 is a random number inside $(0, 1)$, and $J = 2(1 - r_5)$ represents the random jump strength of the rabbit throughout the escaping procedure. The J value changes randomly in each iteration to simulate the nature of rabbit motions.

② Hard besiege

When $|E| < 0.5$ and $r \geq 0.5$, the prey's physical strength is too low to escape pursuit. The expression for this stage is

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)|, \quad (25)$$

③ Soft besiege with progressive rapid dives

When $|E| \geq 0.5$ and $r < 0.5$, the prey still has the energy to escape, and the flock of hawks will form a soft encirclement to hunt the prey. The update expression for this stage is

$$X(t+1) = \begin{cases} Y, & \text{if } f(Y) < f(X(t)), \\ Z, & \text{if } f(Z) < f(X(t)). \end{cases} \quad (26)$$

In the formula, $f(\cdot)$ is the objective function value of the minimization problem, and Y and Z are, respectively,

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \quad (27)$$

$$Z = Y + S \times LF(D), \quad (28)$$

where D is the dimension of the problem; S is a random vector of size $1 \times D$ on $(0, 1)$; and LF is the Levy flight function, calculated as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}},$$

$$\sigma = \frac{\Gamma(1 + \beta) \sin(\pi\beta/2)}{\Gamma\left(\frac{1+\beta}{2}\right) \beta \cdot 2^{(\beta-1)/2}}. \quad (29)$$

In this formula, u and v are both random numbers in $(0, 1)$; β is a default constant, set to $\beta = 1.5$.

④ Hard besiege with progressive rapid dives

When $|E| < 0.5$ and $r < 0.5$, the prey does not have enough energy to escape, and the flock of hawks constructs a tight encirclement to capture the prey. The expression for this strategy is

$$X(t+1) = \begin{cases} Y, & \text{if } F(Y) < F(X(t)) \\ Z, & \text{if } F(Z) < F(X(t)) \end{cases} \quad (30)$$

$$Y = X_{\text{rabbit}}(t) - E|X_{\text{rabbit}}(t) - X_m(t)| \quad (31)$$

$$Z = Y + S \times LF(D). \quad (32)$$

4.2. Multi-Strategy Hybrid Harris Hawks Optimization with Differential Evolution Algorithm

Despite HHO's good performance on multiple benchmarks and engineering optimization problems, there are still some noteworthy drawbacks and limitations. As the problem dimensions increase, the search efficiency of HHO decreases significantly, and it is prone to getting trapped in local optima. In addition, current attack strategy selection is mainly based on a simple combination of random number r and absolute energy value E , lacking adaptability. In actual optimization, different problems may require different strategy combinations and switching timings, and fixed switching rules may not be able to fully utilize the advantages of various strategies.

Therefore, in response to the imbalance between algorithm exploration and development, as well as the tendency of algorithms to fall into local optimal, this study proposes an energy-starvation joint decision function by combining energy and starvation rate. Furthermore, it integrates the differential evolution (DE) algorithm with the HHO algorithm to enhance the global search capability and convergence performance. The improvement in the MSHHODE is as follows.

4.2.1. Hunting Enthusiasm

In basic HHO, considering only energy E leads to a simplistic search pattern and limited adaptability in the algorithm, making it difficult to dynamically adjust the balance between global exploration and local exploitation based on the population's internal state. However, the starvation rate has been described in reference [53], and it has been confirmed that considering the hunger level of the hunter can enhance the performance of the algorithm in solving complex optimization problems and improve the reliability of escaping from local optima. If a Harris hawk is satiated, it has enough energy to fly further to find prey; if it does not eat enough, it will not have enough energy to fly for a long time.

Therefore, this paper comprehensively considers starvation rate $\mathcal{F}$ and energy E as hunting enthusiasm HE to promote the basic HHO algorithm to better balance exploration and exploitation. After calculating the starvation rate $\mathcal{F}$ via Equations (33) and (34), the proposed hunting enthusiasm HE is described in Equation (35):

$$f = h \times \left(\sin\left(\frac{\pi}{2} \times \frac{t}{T}\right) + \cos\left(\frac{\pi}{2} \times \frac{t}{T} \cdot -1\right) \right), \quad (33)$$

$$\mathcal{F} = (2 \times \text{rand}_1 + 1) \times z \times \left(1 - \frac{t}{T}\right) + f, \quad (34)$$

$$HE = \mathcal{F}/E. \quad (35)$$

In this formula, the larger the starvation rate $\mathcal{F}$, the weaker the hunger of the Harris hawk; h is a random number in $(-2, 2)$; and ω is a fixed parameter, taken as $\omega = 2.5$, which is inherited from the setting of the hunger rate formula in the original literature. f is a perturbation term, affected by the current iteration number t ; rand_1 is a random value, taking a value in $(0, 1)$; and z is a random number in $(-1, 1)$, which changes with the number of iterations. When $z < 0$, it indicates that the hawk is hungry; when $z > 0$, it indicates that the hawk is full.

Then, the hunting enthusiasm HE is used to macroscopically control whether the Harris hawk engages in global exploration or localized exploitation. When $|HE| > 1$, the hawk will search for food in different areas, which is the global exploration phase. If $|HE| < 1$, it enters the exploitation phase, and the hawk will search for food nearby.

Then, after entering the partial exploitation stage, the hawk will further select and switch between four besieging strategies based on the rabbit's energy level.

4.2.2. Elite-Assisted Guidance Strategy

In the exploration phase of the optimization algorithm, to guide the population search direction more stably and evenly, an elite group average position guidance strategy is investigated. This strategy does not rely on the accidental advantage of a single elite individual, but instead calculates the geometric center of the current elite group and uses it as the guiding direction, improving the robustness and convergence reliability of the search. Specifically, first, several individuals with optimal fitness are selected according to a defined elite ratio, and the arithmetic mean of their position vectors is calculated to obtain the elite average position. Subsequently, non-elite individuals move closer to this average position with a certain learning rate, while introducing small random perturbations to avoid premature homogenization.

$$X(t+1) = \begin{cases} X_{\text{rand}}(t) - r_1 |X_{\text{rand}}(t) - 2r_2 X(t)|, & q \leq 0.3 \\ (X_{\text{rabbit}}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)), & 0.3 < q \leq 0.6, \\ X(t) + r_5 \left(\frac{1}{N \cdot r_e} \sum_{j=1}^{N \cdot r_e} X_{\text{elite}} - X(t) \right) + r_6 \cdot \text{rand}(D), & q > 0.6 \end{cases} \quad (36)$$

where r_e is the elite ratio, X_{elite} is the location of the elite hawk, and $\text{rand}(D)$ is a D -dimensional random vector. r_5 and r_6 are also random numbers inside $(0, 1)$, which are updated in each iteration.

This guidance method is particularly effective in the middle of the iteration, preventing the entire population from being misled by the local extrema of individual elites, while maintaining the overall progress of the population towards the potential optimal region, balancing the breadth of exploration and the depth of development, and providing a guarantee for the stable convergence of the algorithm in complex multimodal spaces.

4.2.3. Adaptive Differential Evolutionary Strategy

To enhance global search capabilities and population diversity, this paper incorporates the differential evolution (DE) algorithm into the HHO algorithm.

(1) Adaptive triggering mechanism

To intelligently incorporate DE strategies into the HHO algorithm, this paper designs a multi-factor adaptive triggering mechanism. This mechanism comprehensively considers three key factors, namely the algorithm's search state, iteration process, and population diversity, dynamically adjusting the execution probability of the differential evolution

strategy, thus achieving the organic integration and complementary advantages of the two algorithms.

First, an adaptive triggering mechanism based on starvation rate $\mathcal{F}$, current iteration t , and population diversity $\mathcal{D}$ is defined in Equations (37) and (38) to determine the adaptive trigger probability P_{DE} in DE:

$$\mathcal{D} = \frac{1}{N} \sum_{i=1}^N \sqrt{\left[\frac{1}{D} \sum_{k=1}^D \left(\frac{X_{i,k} - \bar{X}_k}{UB_k - LB_k} \right)^2 \right]}, \quad (37)$$

$$P_{DE} = \alpha \times |\mathcal{F}| + \beta \times (1 - t/T) + \gamma \times (1 - \mathcal{D}), \quad (38)$$

where α is the starvation rate weighting coefficient, β is the iteration progress weighting coefficient and γ is for population diversity.

In detail, the starvation rate-dominant term ensures the algorithm responds to the search state—the hungrier the flock (the higher the $|\mathcal{F}|$ value), the higher the probability of DE triggering, guiding the population to conduct broader exploration. In addition, the time decay term follows the explore first, develop later optimization paradigm, assigning a higher DE execution probability in the early stages of iteration and gradually decreasing it as the iteration progresses, achieving a natural transition in search focus. Finally, the diversity compensation term monitors the population distribution in real time—when the population tends to be homogeneous (low diversity $\mathcal{D}$), it automatically increases the DE probability, injecting new individuals to break stagnation and maintain search vitality. These three factors work together to dynamically adjust the intensity of DE intervention, enabling the algorithm to fully utilize the HHO siege strategy while also enhancing global exploration and diversity through DE in a timely manner.

(2) Two-stage DE mechanism

Based on the characteristics of the different stages of the HHO algorithm, this paper designs two targeted differential evolution strategies, which are applied to the exploration and development stages, respectively.

① Exploration Phase: DE/rand/1 Strategy

During the exploration phase ($|HE| \geq 1$), the main objective is to enhance global search capabilities and prevent the algorithm from getting trapped in local optima prematurely. To this end, this paper employs the classic DE/rand/1 mutation strategy, as shown in Equation (39):

$$\mathbf{V}_i(t) = \mathbf{X}_{r_1}(t) + F \cdot (\mathbf{X}_{r_2}(t) - \mathbf{X}_{r_3}(t)) \quad (39)$$

Among these variables, $\mathbf{V}_i(t)$ is the mutation vector, and $\mathbf{X}_{r_1}$, $\mathbf{X}_{r_2}$, and $\mathbf{X}_{r_3}$ are three distinct individuals randomly selected from the current population. F is the scaling factor (taken as $F = 0.5$ in this paper). The completely random nature of this strategy effectively expands the search range and enhances the algorithm's exploration capability in the solution space.

② Exploitation phase: DE/best/1 strategy

During the exploitation phase ($|HE| < 1$), the main task of the algorithm is to perform a fine-grained search in the potential optimal region. To this end, this paper adopts an elite-guided DE/best/1 strategy, as shown in Equation (40):

$$\mathbf{V}_i(t) = \mathbf{X}_{best}(t) + F \cdot (\mathbf{X}_{r_1}(t) - \mathbf{X}_{r_2}(t)). \quad (40)$$

where $\mathbf{X}_{best}(t)$ is the current global optimal solution. This strategy uses the optimal solution as a benchmark and conducts local exploration around the optimal solution through

differential perturbation, which can maintain the convergence trend towards the optimal solution while avoiding getting stuck in local stagnation.

(3) Crossover and selection mechanisms

For both the above two strategies, a binary crossover operation is used to generate the test vectors:

$$U_{i,j}(t) = \begin{cases} V_{i,j}(t), & \text{if } \text{rand}(0,1) \leq CR \text{ or } j = j_{\text{rand}} \\ X_{i,j}(t), & \text{otherwise} \end{cases} \quad (41)$$

where CR is the crossover probability ($CR = 0.7$ in this paper), and j_{rand} is a randomly selected dimension index, ensuring that at least one dimension of the experimental vector comes from the mutation vector.

After generating the experimental vector $\mathbf{U}_i(t)$, a greedy selection mechanism is used to determine whether to update the current individual:

$$\mathbf{X}_i(t+1) = \begin{cases} \mathbf{U}_i(t), & \text{if } f(\mathbf{U}_i(t)) < f(\mathbf{X}_i(t)) \\ \mathbf{X}_i(t), & \text{otherwise} \end{cases} \quad (42)$$

This selection mechanism ensures the monotonically convergent nature of the algorithm, meaning that the optimal fitness of the population will not deteriorate with iteration.

4.3. MSHHODE's Framework

The flowchart for solving the heterogeneous multi-robot task allocation problem in an environment with uncertain task execution duration, latest allowable service time of target equipment, and robot travel speed using the MSHHODE algorithm is summarized as shown in Figure 1, where the gray box represents the improved content of this paper.

Step 1: When the algorithm starts executing, the search parameters and positions of all individuals in the population are first initialized. Then, the fitness value of each individual is calculated, while the initial optimal solution is achieved.

Step 2: At the beginning of each iteration, the diversity $\mathcal{D}$ of the current population is calculated based on the distribution of individuals in the solution space. Then, the triggering probability P_{DE} of the DE strategy in this iteration is also determined according to Equation (38).

Step 3: Based on a predefined elite ratio r_e , individuals with the best fitness are selected to form an elite population.

Step 4: For each individual, calculate their starvation rate $\mathcal{F}$ and hunting enthusiasm value HE , which will guide different solution stages.

Step 5: When the hunting enthusiasm HE is greater than or equal to 1, the exploration stage begins, focusing on a global search. When the hunting enthusiasm HE is less than 1, the development stage begins, focusing on a detailed, localized search.

Step 6: During the exploration phase, the algorithm determines the search strategy based on the adaptive trigger probability P_{DE} :

Case A: Triggering DE Strategy. When the random number r is less than the adaptive trigger probability P_{DE} , the random differential mutation strategy in differential evolution is executed: DE/rand/1 mutation operation, crossover operation, and greedy selection, where the fitnesses of the experimental vector and the original individual are compared, and the better one is retained.

Case B: No DE Strategy Triggered. When the random number r is greater than or equal to the adaptive trigger probability P_{DE} , an elite-assisted guidance strategy is executed. The algorithm randomly selects one of three improved exploration strategies to execute,

including exploration based on random individuals, exploration based on optimal and average positions, and exploration based on elite information.

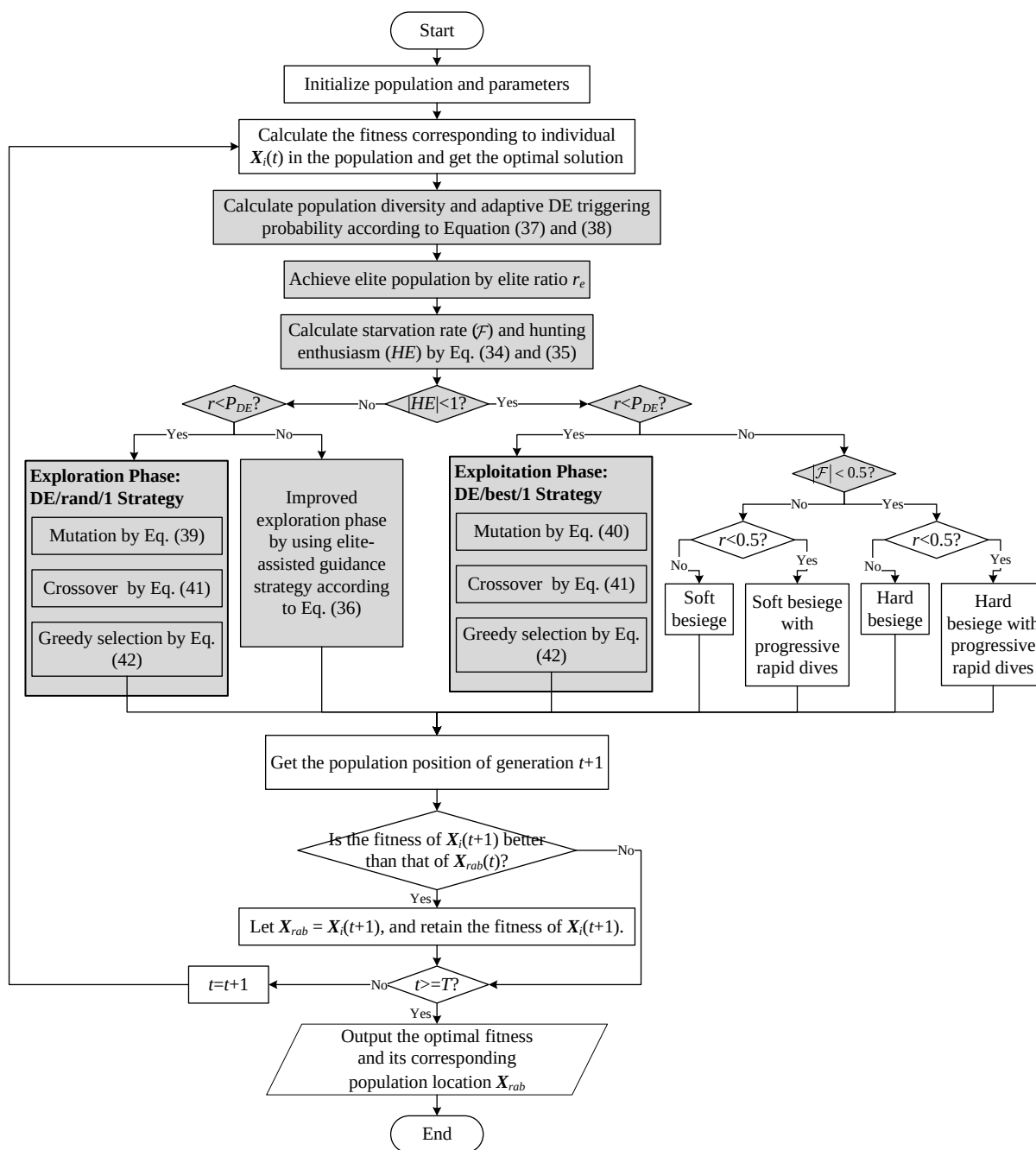


Figure 1. Flowchart of MSHHODE algorithm.

Step 7: During the exploitation phase, strategy selection is also based on adaptive trigger probability P_{DE} :

Case A: Triggering DE Strategy. Execute the DE strategy based on the optimal solution: DE/best/1 mutation operation, crossover operation, and greedy selection.

Case B: Not Triggering DE Strategy. Execute the traditional Harris hawk siege strategy. Based on the probability of random events and the starvation rate $\mathcal{F}$, select from four siege modes: soft besiege, hard besiege, soft besiege with progressive rapid dives, and hard besiege with progressive rapid dives.

Step 8: After updating the positions of all individuals, a new generation of population is formed. The fitness values of all updated individuals are calculated and compared with the historical optimal solution. If a better solution is found, the position and fitness information of the global optimal solution are updated, and the relevant constraint satisfaction status is recorded.

Step 9: Increase the iteration counter and determine if the preset maximum number of iterations has been reached. If not, return to Step 2 to continue execution; if the maximum number of iterations has been reached, proceed to output the final optimal solution.

When the algorithm terminates, it outputs the optimal solution obtained during the entire search process, including the optimal position, the optimal fitness value, and related constraint satisfaction information.

The proposed MSHHODE algorithm intelligently integrates the global exploration capability of differential evolution and the local development capability of Harris hawk optimization through an adaptive mechanism, achieving efficient global optimization search while maintaining population diversity.

4.4. Task Assignment Encoding

In the MSHHODE algorithm, each hawk individual represents a candidate task allocation solution, and the optimal solution is iteratively refined through population search and position updates. To satisfy the temporal sequencing constraints between inspection and maintenance tasks, the constraint that each robot can only execute one task at a time, and the capability matching constraint between robot type and task type in the heterogeneous multi-robot task allocation problem under environments with uncertain task execution duration, latest allowable service time of target equipment, and robot travel speed, this paper adopts a real-vector-based encoding method to establish a mapping relationship between hawk positions and task allocation solutions.

The dimension of the hawk position vector is equal to the total number of tasks, which is twice the number of target equipment units. The integer part $[X]$ of the hawk position in each dimension represents the index of the robot assigned to the corresponding task; tasks sharing the same integer part are assigned to the same robot and will be executed sequentially by that robot. This paper ensures the capability matching between heterogeneous robots and task types by restricting the upper and lower bounds of the integer part of the hawk position in each dimension according to the functional type of each robot. Specifically, the integer part of dimensions corresponding to inspection tasks is restricted to the indices of robots capable of performing inspection tasks, and similarly for maintenance tasks.

For example, as shown in Table 3, the integer part of the hawk position corresponding to each equipment inspection task can only take values from the set of robot indices capable of performing inspection tasks, and the same restriction applies to maintenance tasks. The decimal part $\{X\}$ of the hawk position indicates the execution order of the task within the assigned robot's task list; a smaller decimal part value indicates an earlier execution priority for that task.

After decoding the hawks' positions in the table, the task sequences of robots U_1 , U_2 , and U_3 are obtained as follows: $T_{3,1} \rightarrow T_{2,1}$, $T_{3,2} \rightarrow T_{1,1}$, $T_{1,2} \rightarrow T_{2,2}$.

After obtaining the value of decision variable x_{ij} according to Table 3, it means that the task list for robot execution has been obtained. Then, the start and end times of each task in the form of triangular fuzzy numbers are calculated cumulatively according to Formulas 6 and 8. Then, constraints 13 and 14 are used to determine whether the execution plan meets the temporal and time window constraints of each task. If not, a larger penalty function

is added to the current decision variable plan to encourage MSHHODE to seek a better decision variable plan.

Table 3. The correspondence between Harris hawk position and task allocation result.

Target Equipment	T_1		T_2		T_3	
Task	$T_{1,1}$	$T_{1,2}$	$T_{2,1}$	$T_{2,2}$	$T_{3,1}$	$T_{3,2}$
X	2.6	3.4	1.9	3.6	1.1	2.4
[X]	2	3	1	3	1	2
{X}	0.6	0.4	0.9	0.6	0.1	0.4
Robot	U_2	U_3	U_1	U_3	U_1	U_2

4.5. Computational Complexity Analysis

The time complexity of the standard HHO algorithm is $O(MaxIter \cdot PopSize \cdot Dim)$, where $MaxIter$ is the maximum number of iterations, $PopSize$ is the population size, and Dim is the problem dimension, which in this paper equals $2N$. The improvements introduced in MSHHODE add only constant-order overhead per individual at each iteration: computing the hunting enthusiasm HE requires $O(1)$ operations, and the elite-assisted guidance strategy requires selecting and averaging $PopSize \cdot r_e$ elite individuals, costing $O(PopSize \cdot Dim)$ per iteration for sorting and averaging across the whole population. In addition, the adaptive DE mechanism, including mutation, crossover, and greedy selection, costs $O(Dim)$ per individual. Hence, the overall time complexity of MSHHODE remains $O(MaxIter \cdot PopSize \cdot Dim)$, i.e., the same asymptotic order as the standard HHO, with only a small constant-factor increase due to the additional bookkeeping.

5. Simulation Results

5.1. Parameter Settings and Experimental Environment

The algorithm parameters in this paper are set as follows: maximum number of iterations $MaxIter = 500$; $populationsize = 30$. All algorithms and test programs in this paper are simulated using Matlab R2017a, with the Windows 10 operating system, AMD Ryzen 7 PRO CPU with a clock speed of 1.70GHz, and 16GB of memory.

5.2. Instance Verification of Algorithm Feasibility

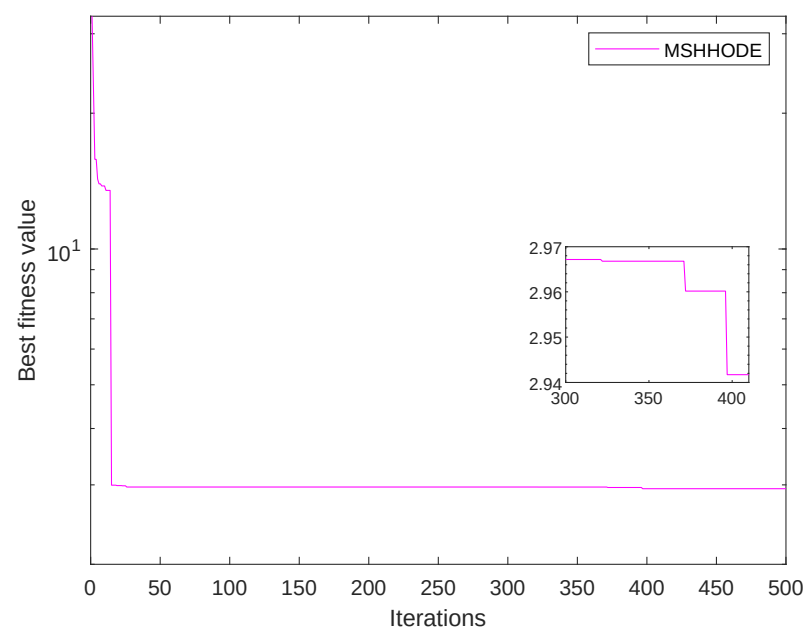
To verify the feasibility of the MSHHODE algorithm in solving the heterogeneous multi-robot task allocation problem under uncertain information, this paper simulates four heterogeneous mobile robots performing inspection and maintenance tasks on eight target equipment units across a large-scale manufacturing facility covering an area of $10,000 \text{ m} \times 10,000 \text{ m}$, which represents a realistic industrial park scenario encompassing multiple production workshops and logistics corridors. It means $M = 4$ and $N = 8$, and sets the decision-maker preference value to $c_r = 0.7$. Tables 4 and 5 are initial information tables of robots and target equipment, respectively. The unit for position is meters (m), the unit for robot travel speed is m/s, and the unit for latest allowable service time and inspection/maintenance duration is seconds (s). The final robot allocation information and task allocation information are shown in Table 6 and Table 7, respectively. Figure 2 shows the fitness change curve for the MSHHODE's solution process.

Table 4. The initial information of the robots.

Robot	Functional Type	Value	Success Prob.	Initial Position	Travel Speed
U_1	Inspection	75	0	(4.204, 90.68)	[3.51, 39, 4.29]
U_2	Inspection & maintenance	166	0.83	(161.49, 95.09)	[3.42, 3.8, 4.18]
U_3	Inspection & maintenance	130	0.91	(190.52, 108.5)	[3.87, 4.3, 4.73]
U_4	Maintenance	87	0.99	(73.847, 187.1)	[3.69, 4.1, 4.51]

Table 5. The initial information of the tasks.

Task	Task Type	Value	Failure Prob.	Initial Position	Latest Allowable Service Time	Execution Duration
$T_{1,1}$	Inspection	61	0.5	(958.44, 621.39)	[2804, 2854, 2904]	[82.8, 92, 101.2]
$T_{1,2}$	Maintenance					[271.8, 302, 332.2]
$T_{2,1}$	Inspection	96	0.6	(811.22, 527.42)	[2500, 2550, 2600]	[81, 90, 99]
$T_{2,2}$	Maintenance					[279, 310, 341]
$T_{3,1}$	Inspection	85	0.4	(540.78, 771.60)	[2865, 2915, 2965]	[84.6, 94, 103.4]
$T_{3,2}$	Maintenance					[256.5, 285, 313.5]
$T_{4,1}$	Inspection	91	0.6	(702.63, 829.32)	[1887, 1937, 1987]	[89.1, 99, 108.9]
$T_{4,2}$	Maintenance					[254.7, 283, 311.3]
$T_{5,1}$	Inspection	70	0.3	(682.94, 962.18)	[2605, 2655, 2705]	[91.8, 102, 112.2]
$T_{5,2}$	Maintenance					[268.2, 298, 327.8]
$T_{6,1}$	Inspection	95	0.4	(830.32, 922.68)	[2048, 2098, 2148]	[90.9, 101, 111.1]
$T_{6,2}$	Maintenance					[281.7, 313, 344.3]
$T_{7,1}$	Inspection	58	0.4	(755.39, 903.08)	[2026, 2076, 2126]	[88.2, 98, 107.8]
$T_{7,2}$	Maintenance					[272.7, 303, 333.3]
$T_{8,1}$	Inspection	77	0.6	(936.96, 574.82)	[1137, 1187, 1237]	[83.7, 93, 102.3]
$T_{8,2}$	Maintenance					[268.2, 298, 327.8]

**Figure 2.** Fitness curve of MSHHODE.

The allocation results shown in Tables 6 and 7 demonstrate that the proposed MSHHODE can allocate all uncertain tasks to robots, where the task list completion time for robots and starting time for tasks are predicted and displayed in the uniform of triangular fuzzy numbers. In addition, Table 6 shows that the time sequence constraint reliability and service time window constraint reliability are both greater than the predefined decision-maker preference of $c_r = 0.7$. This means that the proposed algorithm can achieve feasible solutions for the considered uncertain multi-robot task allocation problem. Furthermore, it can be seen from Figure 2 that the algorithm converges quickly, obtaining a feasible solution at iteration number $t = 14$. Moreover, the inclusion of a strategy to escape local optima allows the MSHHODE algorithm to escape local optima multiple times in the later stages of iteration (iteration number $t > 300$), demonstrating good optimization performance.

Table 6. The allocated information for all tasks.

Task	Allocated Robot	Start Time	Time Sequence Constraint Reliability	Time Window Constraint Reliability
$T_{1,1}$	U_2	[777.91, 858.52, 950.78]	1	0.7013
$T_{1,2}$	U_4	[2461.4, 2727.4, 3008.4]	1	0.7967
$T_{2,1}$	U_2	[418.95, 461.83, 512.05]	1	1
$T_{2,2}$	U_4	[2143.7, 2374.8, 2620.1]	1	1
$T_{3,1}$	U_2	[587.12, 647.71, 717.59]	1	1
$T_{3,2}$	U_4	[1445.3, 1600.6, 1766.5]	1	1
$T_{4,1}$	U_3	[883.78, 979.10, 1080.2]	1	1
$T_{4,2}$	U_4	[1353.2, 1500.3, 1653.9]	1	1
$T_{5,1}$	U_1	[257.48, 283.23, 314.70]	1	1
$T_{5,2}$	U_3	[1353.2, 1500.3, 1653.9]	1	1
$T_{6,1}$	U_1	[384.85, 424.35, 470.37]	1	0.7621
$T_{6,2}$	U_4	[1774.2, 1965.3, 2168.5]	1	1
$T_{7,1}$	U_2	[239.89, 263.88, 293.21]	1	1
$T_{7,2}$	U_3	[943.81, 1046.0, 1153.6]	1	0.9475
$T_{8,1}$	U_3	[436.70, 483.35, 533.75]	1	0.9475
$T_{8,2}$	U_3	[943.81, 1046.0, 1153.6]	1	0.9475

Table 7. The allocated schedules of all robots.

Robot	Task Sequence	Task List Completion Time	Travel Distance
U_1	$T_{5,1} \rightarrow T_{6,1}$	[384.85, 424.36, 470.38]	1257.2
U_2	$T_{7,1} \rightarrow T_{2,1} \rightarrow T_{3,1}$	[777.91, 858.52, 950.78]	2190.8
U_3	$T_{8,1} \rightarrow T_{8,2} \rightarrow T_{7,2} \rightarrow T_{4,1} \rightarrow T_{5,2}$	[1353.2, 1500.3, 1653.9]	1360.3
U_4	$T_{4,2} \rightarrow T_{3,2} \rightarrow T_{6,2} \rightarrow T_{2,2} \rightarrow T_{1,2}$	[2461.4, 2727.4, 3008.4]	1967.6

Table 7 records the sequence of tasks performed by each robot and the distance traveled. Table 6 records the robot ID that performed inspection and maintenance tasks for each target equipment, as well as the credibility of each target equipment satisfying the service time window constraint and the temporal sequence constraint. It can be seen that all tasks can be assigned, and the credibility of all tasks satisfying the above two constraints is greater than the given preference value $c_r = 0.7$. Furthermore, Table 6 uses triangular fuzzy numbers to represent the start time of all tasks. Therefore, this experiment verifies the feasibility of the MSHHODE algorithm in solving task allocation problems.

under conditions of uncertain task time windows, uncertain task execution times, and uncertain robot travel speeds.

5.3. The Impact of Problem Scale on Algorithm Performance

To verify the impact of changes in the number of robots and target equipment on the performance of the MSHHODE algorithm, this paper tests the solution performance of the MSHHODE algorithm under different task load (number of tasks/number of robots) settings, with a fixed number of target equipment units ($N = 10$) and an increase in the number of robots ($M = 4, 6, 8, 10$), and with a fixed number of robots ($M = 4$) and an increase in the amount of target equipment ($N = 8, 10, 12, 16$). The average fitness value change curves during the solution process are shown in Figures 3 and 4.

It should be noted that, from a practical application perspective, the problem size should not be set too large. This paper uses triangular fuzzy numbers to model the task execution time, latest allowable service time window, and robot cruise speed. As shown in Equation (6), the fuzzy time $\tilde{B}T_{ik}$ when the robot arrives at the k -th task position is the sum of the fuzzy execution time and fuzzy travel time of all preceding tasks. As the task sequence grows, the uncertainty interval of the triangular fuzzy number will expand along the task chain, resulting in a continuous widening of the prediction interval for the arrival time of subsequent tasks. If the problem size is too large, this cumulative effect will make it difficult to guarantee the credibility of the service time window constraint. A large number of tasks will be forced to be deleted due to violations of the constraints specified in Equation (13), and the feasibility of the scheme will decrease. Therefore, controlling the problem size within a reasonable range is an inherent requirement to ensure that the fuzzy pre-planning scheme has practical reference value.

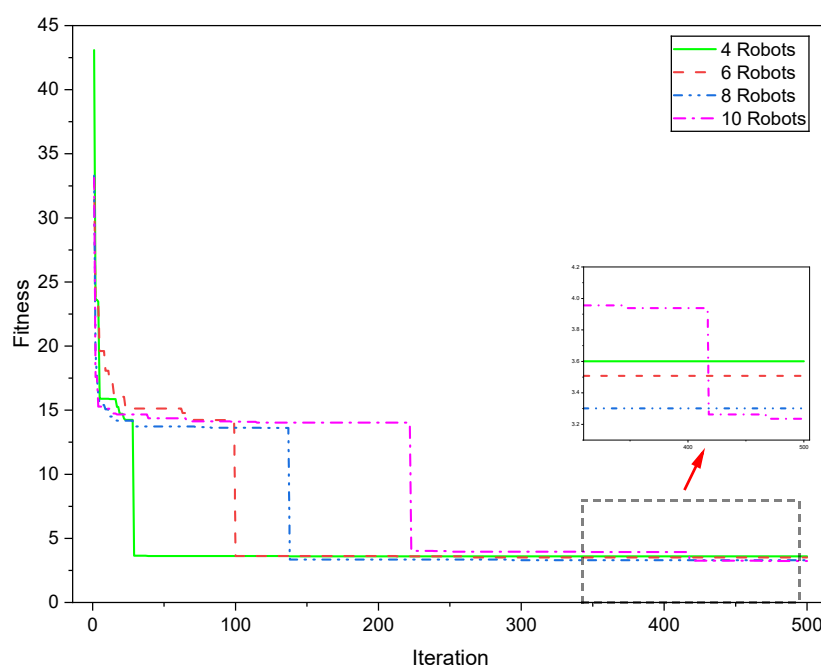


Figure 3. MSHHODE fitness change curve as the number of robots increases.

As shown in Figure 3, the proposed MSHHODE can obtain feasible solutions under all the above scale settings. However, the convergence speed of the fitness value gradually slows down as the number of robots increases. As can be seen from the encoding method proposed, when the amount of target equipment is fixed, the more robots there are, the wider the search domain becomes, leading to a slower solution speed. Furthermore, Figure 3 shows that the fitness value of the final allocation scheme also increases with the

number of robots. This is because when the amount of target equipment is fixed, the total benefit obtained by the robots in performing tasks is not significantly different, but the travel cost and operational risk cost increase continuously with the increase in the number of robots, leading to a continuous increase in the fitness value of the final task allocation scheme.

In addition, as shown in Figure 4, when the number of robots is constant, the convergence speed of the MSHHODE algorithm gradually slows down with the increase in the amount of target equipment. This is because the dimensionality of the individual encoding increases linearly with more target equipment. Therefore, it becomes difficult to find a task allocation scheme that simultaneously satisfies both the temporal and service time window constraints, requiring the algorithm to iterate more times to find feasible solutions. Furthermore, as the number of targets increases, the average number of tasks that each robot needs to perform also increases, leading to an increase in both the task list completion time and distance cost for the robots. Consequently, the final fitness value obtained by MSHHODE also increases with the amount of target equipment.

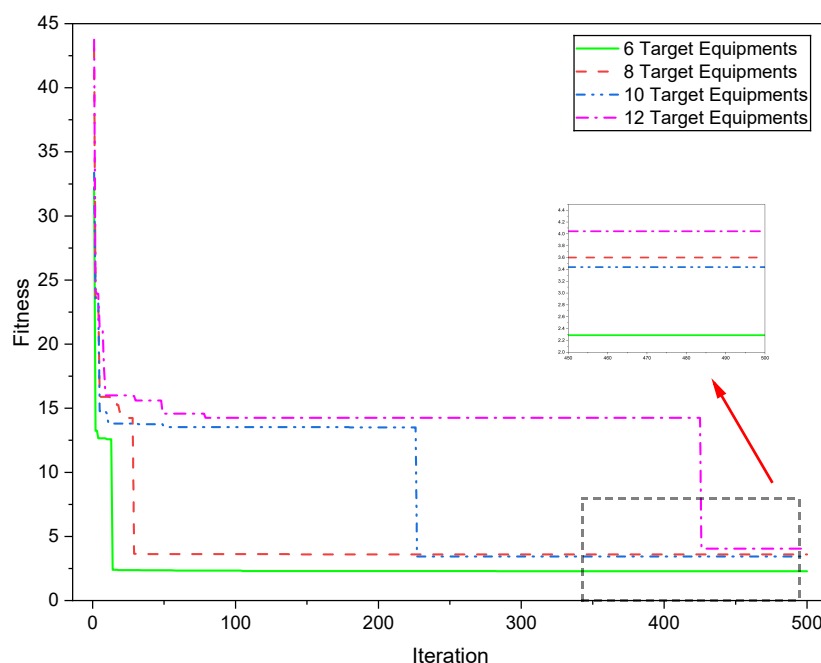


Figure 4. MSHHODE fitness change curve as the amount of target equipment increases.

5.4. Sensitivity Analysis of Model and Algorithm Parameters

This section investigates the sensitivity of MSHHODE to credibility threshold c_r^* , the degree of uncertainty χ , the weighting coefficients $\omega_1, \omega_2, \omega_3$ in the objective function of Equation (12), and the elite ratio r_e in the elite-assisted guidance strategy of Equation (36).

5.4.1. Sensitivity to the Decision-Maker Preferences and Degree of Uncertainty

To verify the impact of decision-maker preference values on the MSHHODE performance, Figures 5–7 statistically illustrate the average fitness value, average time to complete all tasks, and total travel length of all robots for feasible solutions obtained from 50 random instances under four different task load scenarios with three settings: preference values $c_r = 0.3, 0.5, 0.7$.

In addition, since the upper and lower bounds λ_1 and λ_3 of the uncertain information in this paper are determined by λ_2 and the degree of uncertainty χ , in order to verify the impact of the degree of uncertainty of the uncertain information on the performance of the task allocation scheme, Figures 5–7 also show the fitness value, average task completion

time, and total distance of the feasible task allocation schemes that are randomly run 50 times by the MSHHODE algorithm when the degree of uncertainty is low, medium, and high, that is, when χ is 0.1, 0.2, and 0.3, respectively.

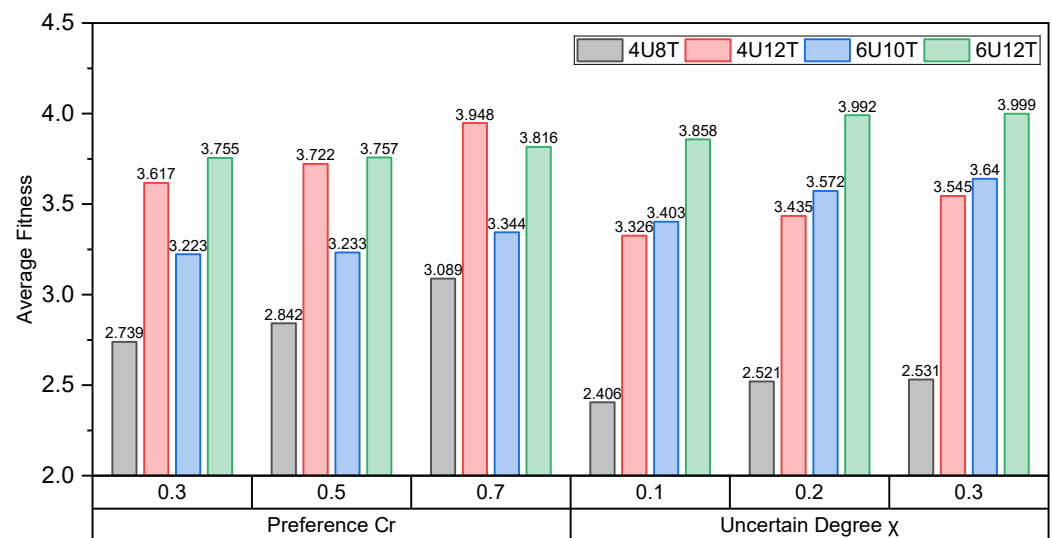


Figure 5. Average fitness of final available solution for the MSHHODE algorithm in four task scenarios under different decision-maker preferences and uncertainties.

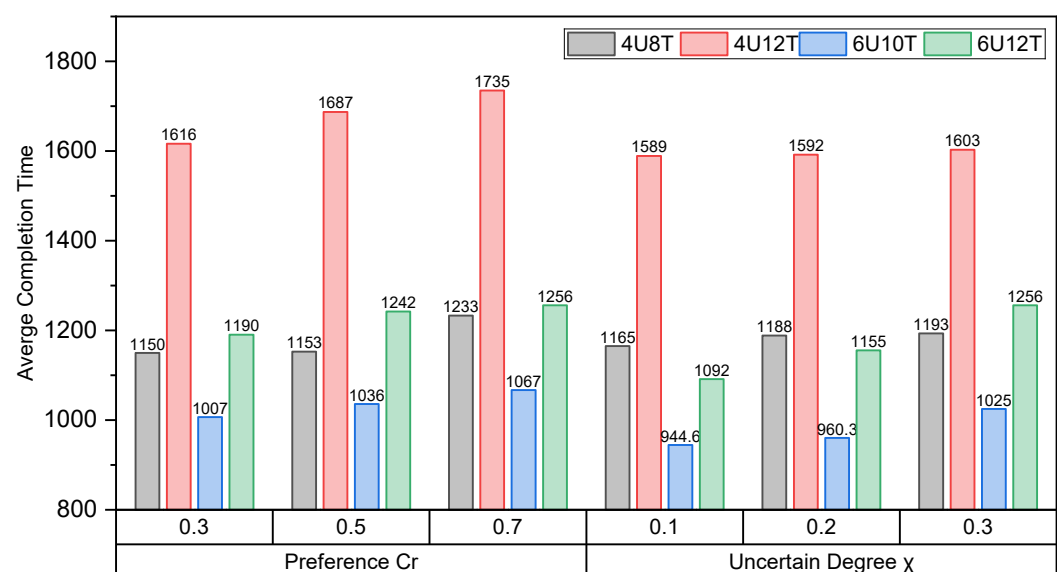


Figure 6. Average completion time of final available solution for the MSHHODE algorithm in four task scenarios under different decision-maker preferences and uncertainties.

As shown in Figures 5 and 6, as the preference value c_r gradually increases, the average robot task list completion time $AVGComTime$ (i.e., time cost) under the three task load settings continues to increase, resulting in a general gradual increase in the fitness value of the allocation schemes. This is because when c_r increases, the decision-maker gradually tends to be conservative. In order to minimize the possibility of violating the target equipment temporal sequencing constraints, the start execution time of target maintenance tasks is delayed, leading to a continuous increase in the robot task list completion time, and consequently, an increase in the fitness value of the obtained task allocation schemes. In addition, due to the increase in the degree of uncertainty χ , to ensure that all temporal sequencing constraints are satisfied, the start time of tasks is delayed, resulting in an

increase in the robot task list completion time, and consequently, an increase in the fitness value of the task allocation schemes.

In addition, as the degree of uncertainty χ and the decision-maker's preference c_r increase, the algorithm tends to assign multiple tasks that are geographically adjacent or the same target to the same robot to avoid violating the time window and sequential constraints, thereby reducing long-distance cooperative maneuvers between robots and leading to a decrease in the total travel path length, which is displayed in Figure 7.

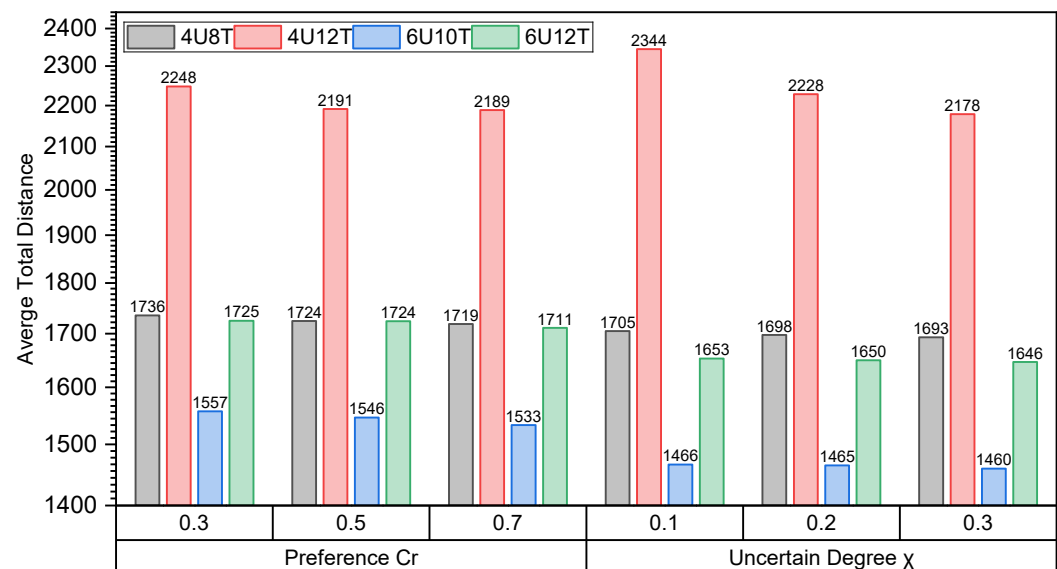


Figure 7. Average total distance of final available solution for the MSHHODE algorithm in four task scenarios under different decision-maker preferences and uncertainties.

5.4.2. Sensitivity to the Weighting Coefficients

Table 8 compares three representative weighting schemes of ω_1 , ω_2 and ω_3 in Equation (12): a balanced setting that is used as the baseline, a risk-averse setting that places greater emphasis on the operational risk cost C_1 , and a travel-cost-oriented setting that places greater emphasis on the travel cost C_2 .

Table 8. Sensitivity analysis of weighting coefficients under scenario 4U8T ($c_r^* = 0.7$, $\chi = 0.2$).

Scenario	ω_1	ω_2	ω_3	AVG Fitness	AVGComTime	AVGDist
Balanced (baseline)	0.34	0.33	0.33	2.5234	1208.31	1699.6
Risk-averse	0.6	0.2	0.2	2.0361	1215.6	1710.3
Travel-cost-oriented	0.2	0.6	0.2	2.7943	1176.82	1689.57

As shown in Table 8, shifting weight toward the operational risk cost increases the average fitness value, as the algorithm tends to favor higher-reliability robots even at the cost of longer travel distances, while shifting weight toward the travel cost reduces *AVGDist* as expected. These results confirm that the weighting scheme in Equation (12) correctly reflects different decision-maker priorities, and that MSHHODE remains stable and consistently able to obtain feasible solutions across all tested weight settings.

5.4.3. Sensitivity to the Elite Ratio

Table 9 further examines the effect of the elite ratio r_e , which controls the size of the elite group used in the elite-assisted guidance strategy, as shown in Equation (36). Three representative values are tested: $r_e = 0.05$, $r_e = 0.20$ (baseline), and $r_e = 0.40$.

As shown in Table 9, both a small ($r_e = 0.05$) and a large ($r_e = 0.40$) elite ratio degrade performance relative to the baseline, reflected in higher *NumNoFeasSol*, *AVG*, and *AVGIter* values. When r_e is too small, the elite-group centroid is derived from too few individuals and may not provide a stable guidance direction; when r_e is too large, the elite group increasingly overlaps with the general population, weakening the guidance effect and slowing convergence. These results confirm that MSHHODE's performance is relatively robust within a reasonable range around the adopted setting $r_e = 0.2$, and provide empirical justification for this parameter choice.

Table 9. Sensitivity analysis of the elite ratio r_e under scenario 4U8T.

r_e	NumNoFeasSol	AVG Fitness	AVGIter
0.05	1	2.5233	19.56
0.20 (baseline)	0	2.4734	15.31
0.40	2	2.514	18.46

5.5. Robust Performance Verification of Allocation Results

Due to the uncertainty of task execution time, target equipment latest execution time, and robot navigation speed, in order to verify the robust performance of the task allocation scheme obtained by the MSHHODE algorithm in the actual task execution process, the stochastic simulation algorithm (SSA) is applied to simulate the actual values of the above uncertain information.

When evaluating the robustness of the task allocation schemes, the first step is to evaluate whether the allocation scheme obtained by MSHHODE can meet the temporal and time window requirements when assigning accurate values to uncertain parameters. If it meets the requirements, the allocation scheme remains unchanged. If it does not meet the requirements, the auction method is used to redistribute tasks that cannot meet the time requirements. Evaluate the robustness of the allocation scheme obtained by MSHHODE by counting the number of times the original scheme needs to be reallocated and the proportion that can be reallocated.

In order to evaluate the robustness performance of the MSHHODE algorithm, it must be evaluated in terms of different task allocation schemes under three uncertainty level settings χ under three different task load settings, as shown in Table 10. Firstly, after obtaining feasible solutions that satisfy the constraints using the MSHHODE algorithm, 50 sets of actual values of uncertain information are randomly generated using the SSA algorithm for each feasible solution. Finally, calculate the proportion of tasks that need to be reassigned in the original task allocation plan *RePer*, and the proportion of successful reassignments *ReSucPer* in this actual situation.

Table 10. Robust performance of MSHHODE under three different uncertainty degrees χ .

Scenario	4U10T			4U12T			6U12T		
Uncertain Degree χ	0.1	0.2	0.3	0.1	0.2	0.3	0.1	0.2	0.3
<i>RePer</i>	43.21%	46.75%	53.71%	48.94%	53.17%	65.39%	51.37%	56.76%	63.53%
<i>ReSucPer</i>	73.25%	81.34%	91.26%	70.73%	78.36%	86.95%	68.31%	74.54%	83.27%

With the increase in task load, more tasks in the task allocation scheme obtained by MSHHODE cannot meet the constraints during actual execution and need to be reassigned more times to ensure the successful completion of the whole mission. And due to this increase in task load, each robot takes longer to complete its task list, which increases the possibility that the temporal sequencing constraints and service time window constraints

of other tasks in the task list will not be able to be met after adding the failed tasks. As a result, the proportion of successful reassignments decreases with the increase in task load.

In addition, when the degree of uncertainty increases, more allocated tasks cannot meet the temporal and service time window constraints in the actual execution process. Therefore, when the task load is constant, the proportion that needs to be reallocated increases with the increase in uncertainty. Meanwhile, the increase in uncertainty means that drones are more likely to utilize redundant time to handle failed tasks; thus, the proportion of successful reassignments increases with the preference of decision-makers.

5.6. Algorithm Performance Comparison

To verify the superiority of the proposed algorithm, this paper considers five different settings (4U8T, 4U10T, 4U12T, 6U10T, 6U12T). MSHHODE and five other classic algorithms, including Grey Wolf Optimizer (GWO), Improved Grey Wolf Optimizer (IGWO), Dream Optimization Algorithm (DOA) [54], Logistic-Gauss Circle optimizer (LGC) [55], and the standard Harris Hawks Optimization (HHO), are independently run 50 times under each setting.

First, Table 11 presents the number of times a feasible solution cannot be found (*NumNoFeasSol*) over the 50 random task-allocation instances, directly indicating the solving capability of the six algorithms. *BST*, *AVG*, *WST*, *STD*, and *AVGIter* represent the best, average, worst, standard deviation, and average number of iterations required to obtain a feasible solution, respectively, after 50 random runs of each algorithm. *AVGDis* and *AVGComTime* denote the average total distance traveled by all robots and the average task-list completion time, respectively. Since the task-list completion time for each robot is expressed as a triangular fuzzy number, the most probable estimate (the center value of the triangular fuzzy number) is used to calculate the average completion time. Optimal results are highlighted in bold.

To further verify that the performance improvement in MSHHODE is statistically significant rather than due to random variation, a Wilcoxon rank-sum test is conducted between MSHHODE and each benchmark algorithm within each problem-scale scenario, based on the 50 independent runs reported in Table 11. In addition, a Friedman test is performed across all five scenarios to examine the overall ranking of the six algorithms in terms of average fitness, as displayed in Table 12.

Furthermore, to illustrate the convergence performance of all six algorithms, this paper selects from the 50 random runs under each setting (excluding instances where no feasible solution is found) and plots the average convergence curves for the six algorithms. These curves are shown in Figures 8–12 for the five different problem-scale settings.

According to the *NumNoFeasSol* metric in Table 11, it can be observed that the proposed MSHHODE algorithm can obtain feasible solutions the most times across all settings across 50 random instances. Furthermore, its *BST* ranks first in Settings 2, 3 and 4, and ranks second in Settings 1 and 5. In Setting 3, its *BST* value significantly outperforms the other algorithms. This directly demonstrates that the algorithm possesses stronger global exploration capabilities and can effectively avoid falling into local optimum traps. Additionally, its *AVG* values in all settings are the lowest among all algorithms, indicating that the algorithm exhibits optimal average performance under different initial conditions and high reliability. Furthermore, its *WST* value is much lower than that of other algorithms, and its *STD* value is also lower than that of the other five algorithms. This means that the quality fluctuation of the MSHHODE solution is small and it has strong robustness. In addition, it can be observed that the basic HHO algorithm has the worst solution performance. It fails to find a feasible solution the most times, and the *BST* and *WST* of the solution are worse than the other algorithms.

Table 11. Comparison of task allocation results for six algorithms.

Scenario	Algorithms	NumNoFeasSol	BST	AVG	WST	STD	AVGDist	AVGComTime	AVGIter	p-Value (Wilcoxon)
No 1: 4U8T	GWO	0	2.8332	2.9006	2.9831	0.3461	1751.004	1206.98	32.1	2.1×10^{-3}
	IGWO	0	2.8321	3.0886	3.0886	0.1284	1776.518	1232.89	51.54	7.6×10^{-4}
	DOA	0	2.8143	2.8796	2.9592	0.0342	1754.013	1204.66	26.9	5.2×10^{-3}
	HHO	1	2.8591	3.1335	3.1335	0.1122	1778.538	1162.49	68.2	3.4×10^{-8}
	LGC	0	2.8874	2.9134	3.0274	0.0607	1792.61	1197.31	34.95	8.5×10^{-3}
	MSHHODE	0	2.8251	2.8721	2.9348	0.0259	1755.942	1183.27	23.2	/
No 2: 4U10T	GWO	4	3.3918	4.1771	4.6743	0.2520	1996.714	1417.19	143.32	4.5×10^{-6}
	IGWO	1	3.3587	3.5945	4.3572	0.1120	2017.914	1428.72	107.98	8.1×10^{-5}
	DOA	2	3.3598	3.7753	4.7753	0.1727	2037.594	1410.76	123.88	3.2×10^{-4}
	HHO	11	3.4272	4.9358	5.9358	0.0425	2007.954	1435.08	288.48	6.9×10^{-9}
	LGC	2	3.3682	3.8617	4.4341	0.1307	2016.10	1391.39	118.34	1.5×10^{-3}
	MSHHODE	0	3.3581	3.4122	3.4747	0.0264	2034.544	1380.54	74.32	/
No 3: 4U12T	GWO	15	3.9846	4.0234	4.4733	0.0454	2260.684	1600.30	295	2×10^{-7}
	IGWO	12	3.9743	4.0119	4.8685	0.0493	2259.394	1593.10	145.57	8×10^{-6}
	DOA	5	3.8691	3.9568	5.585	0.0896	2265.614	1622.89	192.28	4×10^{-5}
	HHO	17	3.8332	4.7609	5.1991	0.1345	2236.464	1552.61	313	1×10^{-10}
	LGC	11	3.4751	3.8167	4.4319	0.0612	2301.347	1590.1	183.7	7×10^{-4}
	MSHHODE	5	2.9857	3.8288	4.2417	0.0447	2227.746	1606.41	155.14	/
No 4: 6U10T	GWO	10	3.3062	3.3738	3.8852	0.0508	1584.549	1074.99	221.5	5×10^{-4}
	IGWO	9	3.3243	3.4273	3.795	0.0489	1586.089	1019.03	115	8×10^{-4}
	DOA	3	3.2816	3.4488	3.5067	0.0491	1573.291	987.91	441.5	2×10^{-3}
	HHO	12	3.3882	3.4315	4.4181	0.0466	1608.652	959.91	417.5	3×10^{-6}
	LGC	7	3.4314	3.4921	3.9215	0.0410	1610.646	962.6	193.6	4×10^{-3}
	MSHHODE	3	3.2507	3.2856	3.6785	0.0369	1584.057	948.01	94.5	/
No 5: 6U12T	GWO	12	4.0387	4.1347	4.1834	0.0434	1633.300	1092.40	370.5	0.012
	IGWO	9	4.0209	4.1163	4.2125	0.0484	1645.500	1116.50	300.23	0.008
	DOA	8	3.2321	3.9824	4.5924	0.3289	1661.800	1064.50	287.06	0.004
	HHO	14	4.0161	4.1001	4.1643	0.0598	1687.700	1075.30	245	0.0008
	LGC	12	3.7195	4.0694	4.3018	0.0761	1638.419	1106.3	312.6	0.017
	MSHHODE	4	3.3479	3.9809	4.1283	0.0514	1626.600	1075.20	206.23	/

The Optimal result is highlighted in bold

Table 12. Friedman test results across five problem-scale scenarios based on average fitness values in Table 11.

Algorithm	No. 1	No. 2	No. 3	No. 4	No. 5	Mean Rank
GWO	3	5	5	2	6	4.2
IGWO	5	2	4	3	5	3.8
DOA	2	3	3	5	2	3.0
HHO	6	6	6	4	4	5.2
LGC	4	4	1	6	3	3.6
MSHHODE	1	1	2	1	1	1.2

Friedman $\chi^2 = 12.89$, $p = 0.024$.

In terms of the reliability and stability of algorithm convergence, the *AVGIter* metric of MSHHODE is significantly lower than that of the comparative algorithms. For instance, in Settings 2 and 4, it is only 72.32/94.5 iterations, whereas DOA and HHO reach as high as 123.88/441.5 and 288.48/417.5 iterations, respectively. This implies that MSHHODE can enter the feasible solution region more rapidly, significantly enhancing the early convergence efficiency.

MSHHODE also demonstrates superior performance in terms of task execution efficiency and path economy. In four different task settings, its optimized *AVGDist* performs exceptionally well across multiple scenarios, particularly in Setting 3, where its path length is the shortest among all algorithms. This indicates that the algorithm effectively plans a more economical travel trajectory, saving energy and travel time. Simultaneously, its corresponding *AVGComTime* is competitive in multiple settings, such as having the shortest time in Setting 2, Setting 4, and Setting 5. This demonstrates that MSHHODE not only finds shorter paths but also optimizes time costs, achieving a good balance between task efficiency and energy economy.

As shown in Table 11, MSHHODE achieves the lowest AVG fitness value in all five scenarios, and the Wilcoxon rank-sum test confirms that this improvement is statistically significant ($p < 0.05$) over most benchmark algorithms, particularly HHO (p ranging from 3.4×10^{-8} to 8×10^{-4} across scenarios). To assess the overall performance across all tested problem scales, a Friedman test was further conducted based on the AVG values in Table 11. As shown in Table 12, MSHHODE attains the best mean rank among the six algorithms, and the Friedman test confirms this overall advantage is statistically significant where $\chi^2 = 12.89$, $p = 0.024$, demonstrating that MSHHODE consistently outperforms the benchmark algorithms across different problem scales.

As shown in Figures 8–12, the proposed MSHHODE algorithm exhibits faster convergence speed than the other five algorithms in the early stages, and its final average fitness value is significantly lower. Furthermore, it can escape local optima multiple times in the later stages of iteration ($t > 450$), demonstrating good optimization capabilities. The DOA algorithm ranks second in convergence performance, outperforming the other three algorithms in both convergence speed and optimal solution. The basic HHO algorithm, however, shows the worst convergence performance. This fully demonstrates that the MSHHODE algorithm proposed outperforms the other five algorithms in solving the heterogeneous multi-robot task allocation problem with uncertain task execution durations, uncertain target equipment latest execution times, and uncertain robot travel speeds. Moreover, as shown in Table 3, the proposed encoding method reveals that the algorithm's difficulty increases with the increase in task load, requiring all six algorithms to undergo more iterations to further seek optimal allocation results.

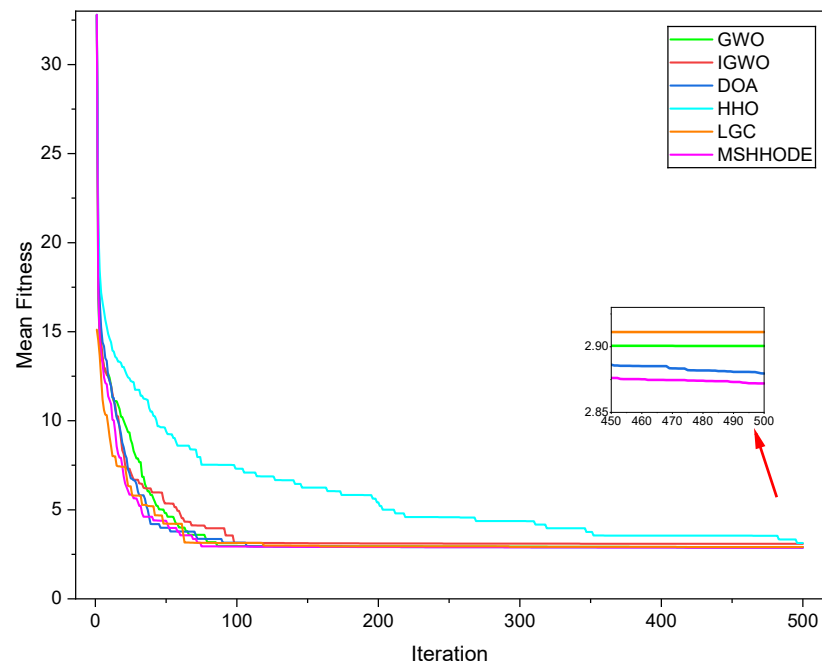


Figure 8. Average fitness curves of six algorithms with 4 robots and 8 target equipment units in 50 random scenarios.

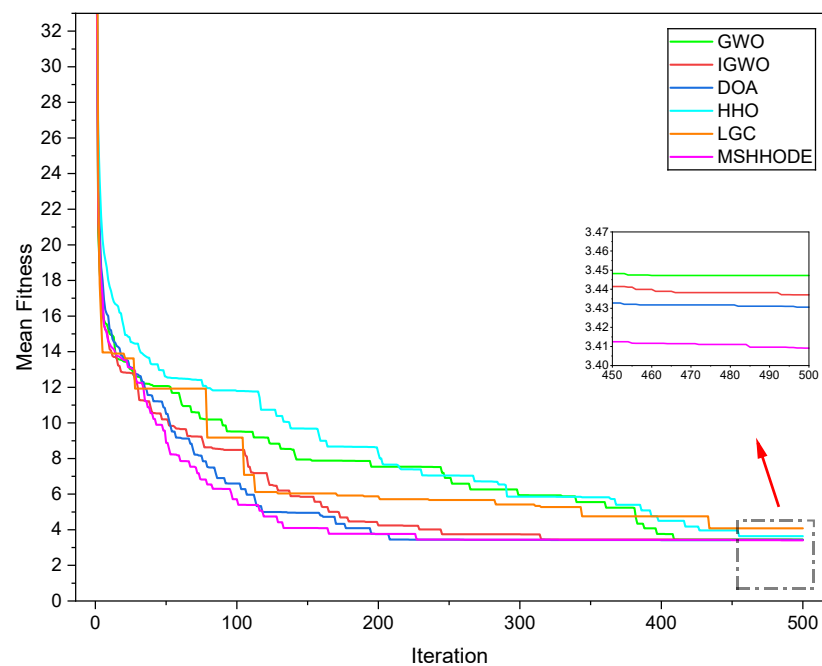


Figure 9. Average fitness curves of six algorithms with 4 robots and 10 target equipment units in 50 random scenarios.

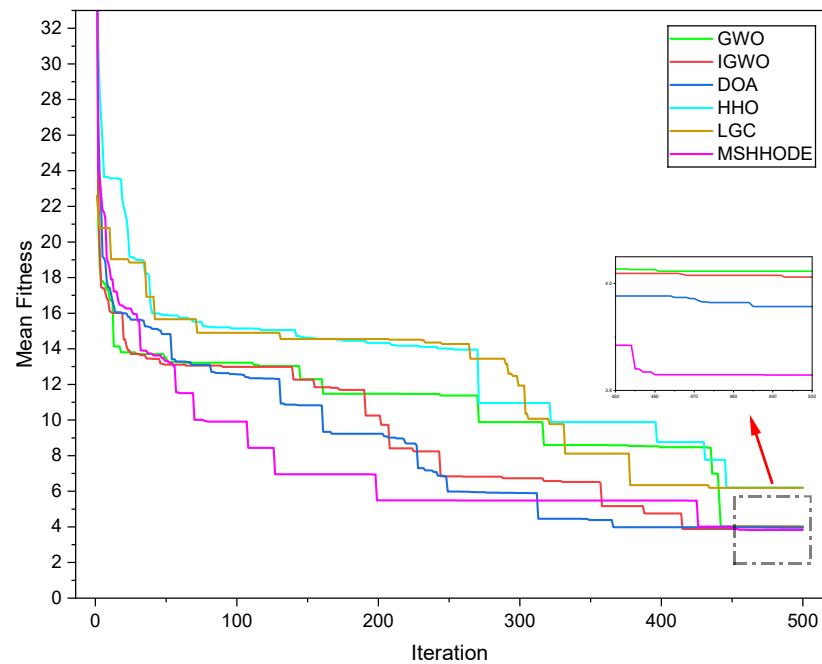


Figure 10. Average fitness curves of six algorithms with 4 robots and 12 target equipment units in 50 random scenarios.

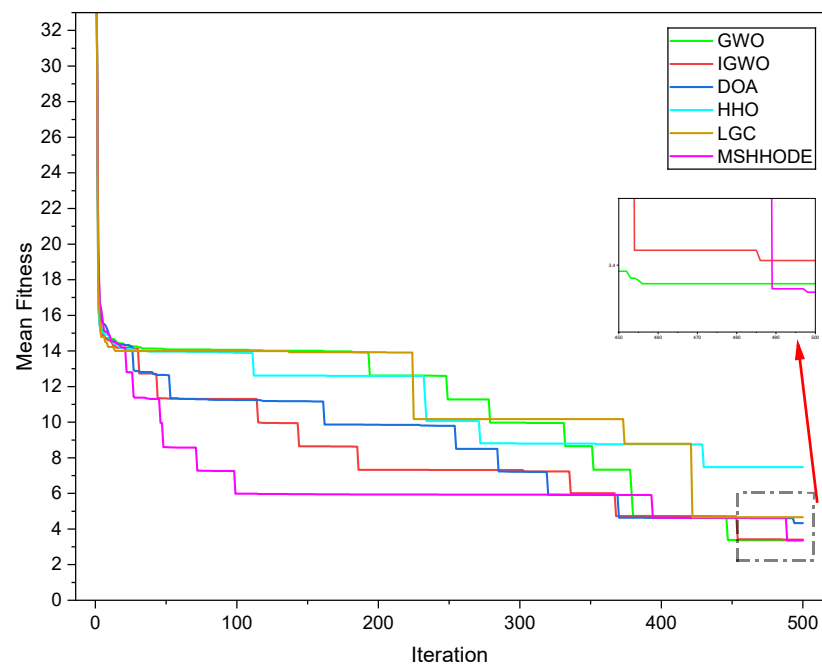


Figure 11. Average fitness curves of six algorithms with 6 robots and 10 target equipment units in 50 random scenarios.

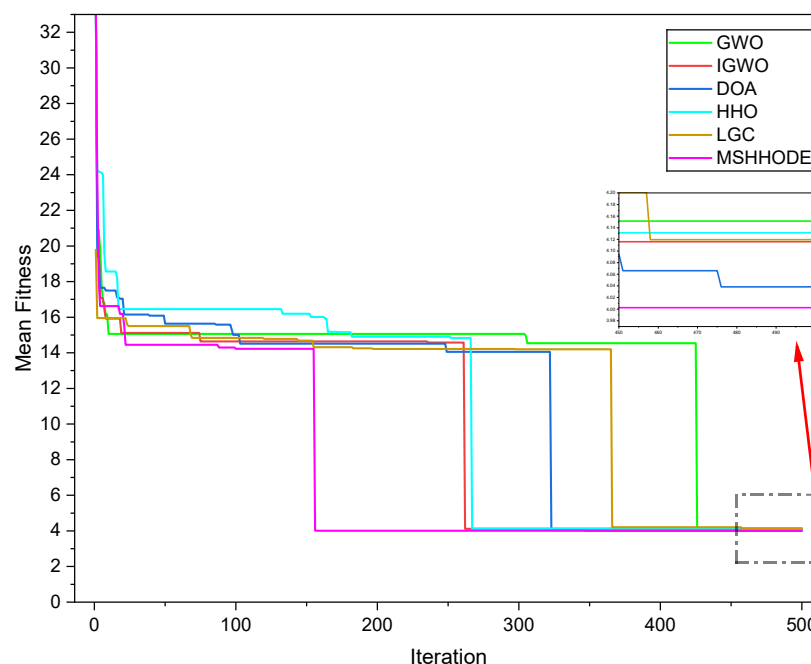


Figure 12. Average fitness curves of six algorithms with 6 robots and 12 target equipment units in 50 random scenarios.

6. Conclusions

This paper addresses the heterogeneous multi-robot task allocation problem in smart manufacturing workshops under uncertain operational conditions, including uncertain task execution durations, latest allowable service times of target equipment, and robot travel speeds. A fuzzy chance-constrained programming model with triangular fuzzy numbers is constructed to characterize these uncertainties, circumventing the need for precise probability distribution estimation and offering greater modeling flexibility for data-scarce workshop environments. Then, to efficiently solve this NP-hard combinatorial optimization problem, a Multi-Strategy Hybrid Harris Hawks Optimization with Differential Evolution algorithm (MSHHODE) is proposed, which incorporates three key improvements: (1) a hunting enthusiasm factor for dynamic phase switching between global exploration and local exploitation; (2) an elite-assisted guidance strategy for robust and stable population convergence; and (3) an adaptive differential evolution strategy to enhance global search capabilities and maintain population diversity throughout the optimization process. The simulation results demonstrate that MSHHODE outperforms five other benchmark algorithms across multiple problem-scale scenarios. The algorithm achieves faster convergence speed, higher solution quality, stronger robustness, and better stability. Particularly in large-scale and complex task scenarios, MSHHODE maintains solution feasibility with fewer failed runs and efficiently escapes local optima through its adaptive mechanisms.

In conclusion, MSHHODE provides an effective and reliable optimization framework for heterogeneous multi-robot task allocation under uncertainty in smart manufacturing environments, offering promising potential for real-world industrial applications such as collaborative equipment inspection and maintenance scheduling in dynamic workshop settings. Future work can extend the uncertainty parameters as trapezoidal fuzzy numbers with flat-topped possible value ranges for parameters or intuitive fuzzy sets used to capture expert hesitation, as well as systematic comparisons of their impact on solution feasibility and computational cost.

Author Contributions: Conceptualization and methodology, M.Y.; software and validation, Z.Z.; investigation, X.Z.; writing—review and editing, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Xihang University's Low-Altitude Economy Special Project grant number 2025PI01 and the Natural Science Basic Research Project of Shaanxi Province grant number 2025JC-YBMS-584.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

- Chen, H.; Li, S.; Fan, J.; Duan, A.; Yang, C.; Navarro-Alarcon, D.; Zheng, P. Human-in-the-loop robot learning for smart manufacturing: A human-centric perspective. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 11062–11086. <https://doi.org/10.1109/TASE.2025.3528051>.
- Shaikat, F.B.; Islam, R.; Happy, A.T.; Faysal, S.A. Optimization of production scheduling in smart manufacturing environments using machine learning algorithms. *Lett. High Energy Phys.* **2025**, *12*, 1–15.
- Wang, G.; Zhang, C.; Liu, S.; Zhao, Y.; Zhang, Y.; Wang, L. Multi-robot collaborative manufacturing driven by digital twins: Advancements, challenges, and future directions. *J. Manuf. Syst.* **2025**, *82*, 333–361. <https://doi.org/10.1016/j.jmsy.2025.06.014>.
- Zhang, T.; Wang, N.; Yang, Y.; Wang, Z. A generalised system for multi-mobile robot cooperation in smart manufacturing. *Robot. Comput.-Integr. Manuf.* **2026**, *98*, 103139. <https://doi.org/10.1016/j.rcim.2025.103139>.
- Wang, G.; Zhang, G.; Zhou, X.; Zhang, Y. Distributed multi-robot task dynamic allocation in digital-twin factory towards industry 5.0. *Int. J. Prod. Res.* **2025**, *64*, 5462–5485. <https://doi.org/10.1080/00207543.2025.2499866>.
- Yang, H.; Xu, W.; Pham, D.T.; Qi, L.; Ba, M. Decentralised task planning and motion coordination for scalable multi-robot collaborative manufacturing. *Robot. Comput.-Integr. Manuf.* **2026**, *100*, 103255. <https://doi.org/10.1016/j.rcim.2026.103255>.
- Chu, D.; Yu, S.; Ling, Y.; Zhao, Y.; Zhang, J. A game-theoretic and multimodal interaction framework for collaborative robots in smart manufacturing. *Decis. Mak. Appl. Manag. Eng.* **2025**, *8*, 36–52. <https://doi.org/10.31181/dmame7120241433>.
- Qiao, P. Optimization of multirobot communication and collaborative control in intelligent manufacturing. In *Proceedings of the Second International Conference on Communication, Information, and Digital Technologies (CIDT 2025)*; SPIE: San Francisco, CA, USA, 2026; Volume 14064, pp. 61–66. <https://doi.org/10.1117/12.3091670>.
- Alqefari, S.; Menai, M.E.B. Multi-UAV task assignment in dynamic environments: Current trends and future directions. *Drones* **2025**, *9*, 75. <https://doi.org/10.3390/drones9010075>.
- Ghauri, S.A.; Sarfraz, M.; Qamar, R.A.; Sohail, M.F.; Khan, S.A. A review of multi-uav task allocation algorithms for a search and rescue scenario. *J. Sens. Actuator Netw.* **2024**, *13*, 47. <https://doi.org/10.3390/jsan13050047>.
- Turgay, S.; Başar, E.E.; Çalışan, M.; Geçkil, M.F.; Baydaş, M.; Stević, Ž. Enhancing decision quality in smart manufacturing: Uncertainty-aware evaluation of edge-cloud architectures with T-spherical hesitant fuzzy rough sets. *Int. J. Comput. Methods Exp. Meas* **2025**, *13*, 211–225. <https://doi.org/10.18280/ijcmem.130201>.
- Fu, B.; Bi, M.; Umeda, S.; Nakano, T.; Nonaka, Y.; Zhou, Q.; Matsui, T.; Tilbury, D.M.; Barton, K. Digital Twin-Based Smart Manufacturing: Dynamic Line Reconfiguration for Disturbance Handling. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 14892–14905. <https://doi.org/10.1109/TASE.2025.3563320>.
- Li, K.; Huang, J.; Zhang, C.; Li, X.; Gao, L.; Shen, W. A GA-CP method for multi-robot collaborative flexible job shop scheduling with time-window constraints. *J. Intell. Manuf.* **2026**, 1–39. <https://doi.org/10.1007/s10845-026-02848-w>.
- Kuo, H.A.; Hong, T.Y.; Chien, C.F. A deep reinforcement learning based digital twin framework for resilient production planning under demand uncertainty and an empirical study in semiconductor wafer fabrication. *Comput. Ind. Eng.* **2025**, *208*, 111389. <https://doi.org/10.1016/j.cie.2025.111389>.
- Durugbo, C.M. Uncertainties of supply chain transition and integration for Industry 5.0: A review of operations management literature. *Int. J. Prod. Res.* **2025**, *63*, 6915–6937. <https://doi.org/10.1080/00207543.2025.2489039>.
- Ostovari, A.; Benyoucef, L.; Haddou Benderbal, H.; Delorme, X. Enhancing scalable reconfigurable manufacturing systems through robust optimisation: Energy efficiency and cost minimisation under uncertainty. *Int. J. Prod. Res.* **2025**, *63*, 3064–3089. <https://doi.org/10.1080/00207543.2024.2445710>.
- Jian, J.; Chen, Y.; Li, Q.; Li, H.; Zheng, X.; Han, C. Decision-making method of multi-UAV cooperate air combat under uncertain environment. *IEEE J. Miniaturiz. Air Space Syst.* **2024**, *5*, 138–148. <https://doi.org/10.1109/JMASS.2024.3378726>.
- Antony Jose, S.; Tonner, A.; Feliciano, M.; Roy, T.; Shackleford, A.; Menezes, P.L. Smart manufacturing for high-performance materials: Advances, challenges, and future directions. *Materials* **2025**, *18*, 2255. <https://doi.org/10.3390/ma18102255>.

19. Florescu, A.; Barabas, S.A. Modeling and simulation of a flexible manufacturing system—A basic component of industry 4.0. *Appl. Sci.* **2020**, *10*, 8300. <https://doi.org/10.3390/app10228300>.
20. Sufian, A.T.; Abdullah, B.M.; Miller, O.J. Smart manufacturing application in precision manufacturing. *Appl. Sci.* **2025**, *15*, 915. <https://doi.org/10.3390/app15020915>.
21. Ahangar, M.N.; Farhat, Z.; Sivanathan, A. AI trustworthiness in manufacturing: Challenges, toolkits, and the path to industry 5.0. *Sensors* **2025**, *25*, 4357. <https://doi.org/10.3390/s25144357>.
22. Mata, O.; Ponce, P.; Perez, C.; Ramirez, M.; Anthony, B.; Russel, B.; Apte, P.; MacCleery, B.; Molina, A. Digital twin designs with generative AI: Crafting a comprehensive framework for manufacturing systems. *J. Intell. Manuf.* **2026**, *37*, 1049–1072. <https://doi.org/10.1007/s10845-025-02583-8>.
23. Virmani, N.; Singh, R.K.; Gupta, S. Empowering Flexible Manufacturing Systems through Generative-AI: A Pathway to Intelligent and Adaptive Production Systems. *Glob. J. Flex. Syst. Manag.* **2026**, 1–23. <https://doi.org/10.1007/s40171-026-00501-2>.
24. Puthanveetil Madathil, A.; Luo, X.; Liu, Q.; Walker, C.; Madarkar, R.; Qin, Y. A review of explainable artificial intelligence in smart manufacturing. *Int. J. Prod. Res.* **2025**, *63*, 8654–8697. <https://doi.org/10.1080/00207543.2025.2513574>.
25. Li, D.; Liu, S.; Wang, B.; Yu, C.; Zheng, P.; Li, W. Trustworthy AI for human-centric smart manufacturing: A survey. *J. Manuf. Syst.* **2025**, *78*, 308–327. <https://doi.org/10.1016/j.jmsy.2024.11.020>.
26. Atanassov, K.; Andonov, V. Generalized nets and intuitionistic fuzzy pairs as tools for modelling of flexible manufacturing systems. *Notes Intuition. Fuzzy Sets* **2020**, *26*, 40–69. <https://doi.org/10.7546/nifs.2020.26.2.40-69>.
27. Praveen, R.; Aida, R.; Rambhatla, A.K.; Trakroo, K.; Maran, M.; Sharma, S. Hybrid Fuzzy Logic-Genetic Algorithm Framework for Optimized Supply Chain Management in Smart Manufacturing. In *Proceedings of the 2025 10th International Conference on Communication and Electronics Systems (ICCES)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1487–1492. <https://doi.org/10.1109/ICCES67310.2025.11336907>.
28. Basciftci, B.; Ahmed, S.; Gebraeel, N. Adaptive two-stage stochastic programming with an analysis on capacity expansion planning problem. *Manuf. Serv. Oper. Manag.* **2024**, *26*, 2121–2141. <https://doi.org/10.48550/arXiv.1906.03513>.
29. Gago-Carro, I.; Aldasoro, U.; Ceberio, J.; Merino, M. A stochastic programming model for ambulance (re) location–allocation under equitable coverage and multi-interval response time. *Expert Syst. Appl.* **2024**, *249*, 123665. <https://doi.org/10.1016/j.eswa.2024.123665>.
30. Gong, H.; Yang, X. A Two-stage Stochastic Programming for the Integrated Emergency Mobility Facility Allocation and Road Network Design Under Uncertainty. *Netw. Spat. Econ.* **2025**, *25*, 445. <https://doi.org/10.21203/rs.3.rs-3839288/v1>.
31. Zhang, Y.; Gong, Y.; Fan, L.; Wang, Y.; Han, Z.; Guo, Y. Quantum-assisted online task offloading and resource allocation in mec-enabled satellite-aerial-terrestrial integrated networks. *IEEE Trans. Mob. Comput.* **2025**, *24*, 3878–3889. <https://doi.org/10.1109/TMC.2024.3519060>.
32. Sengupta, R.; Nagi, R.; Sreenivas, R.S. Robust Task Allocations by Distributing the Risk Among Agents: Theory and Algorithms. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 6475–6491. <https://doi.org/10.1109/TASE.2024.3446456>.
33. Fariman, S.K.; Danesh, K.; Pourtalebiyan, M.; Fakhri, Z.; Motallebi, A.; Fozooni, A. A robust optimization model for multi-objective blood supply chain network considering scenario analysis under uncertainty: A multi-objective approach. *Sci. Rep.* **2024**, *14*, 9452. <https://doi.org/10.1038/s41598-024-57521-0>.
34. Wang, C.; Miao, L.; Zhang, C.; Wu, T.; Liang, Z. Robust optimization for the integrated berth allocation and quay crane assignment problem. *Nav. Res. Logist. (NRL)* **2024**, *71*, 452–476. [https://doi.org/10.1002/nav.22159DigitalObjectIdentifier\(DOI\)](https://doi.org/10.1002/nav.22159DigitalObjectIdentifier(DOI)).
35. Fathollahi-Fard, A.M.; Woodward, L.; Akhrif, O. A scenario-based robust optimization model for the sustainable distributed permutation flow-shop scheduling problem. *Ann. Oper. Res.* **2024**, 1–42. <https://doi.org/10.1007/s10479-024-05940-7>.
36. Mishra, S. Fuzzy assignment model with fuzzy approach. In *Advances in Networks, Intelligence and Computing*; CRC Press: Boca Raton, FL, USA, 2024; pp. 643–649.
37. Ren, Y.; Chen, Q.; Lau, Y.y.; Dulebenets, M.A.; Li, B.; Li, M. A multi-objective fuzzy programming model for port tugboat scheduling based on the Stackelberg game. *Sci. Rep.* **2024**, *14*, 25057. <https://doi.org/10.1038/s41598-024-76898-6>.
38. Huang, H.; Jiang, Z.; Yan, T.; Bai, Y. Dynamic task allocation for heterogeneous multi-UAVs in uncertain environments based on 4DI-GWO algorithm. *Drones* **2024**, *8*, 236. <https://doi.org/10.3390/drones8060236>.
39. Bhoi, S.B.; Dhodiya, J.M. Multi-objective faculty course assignment problem based on the double parametric form of fuzzy preferences. *Oper. Res. Decis.* **2024**, *34*, 1–16. <https://doi.org/10.37190/ord240201>.
40. Liang, D.; Li, Y.; Li, Y.; Xin, J. Dynamic selection and order allocation of resilient suppliers based on improved fuzzy multi-criteria decision method. *J. Chin. Inst. Eng.* **2024**, *47*, 442–455. <https://doi.org/10.1080/02533839.2024.2334201>.
41. Huang, H.Z.; Li, H.; Shi, Y.; Huang, T.; Yang, Z.; He, L.; Liu, Y.; Jiang, C.; Li, Y.F.; Beer, M.; et al. Theory and application of possibility and evidence in reliability analysis and design optimization. *J. Reliab. Sci. Eng.* **2025**, *1*, 015007. <https://doi.org/10.1088/3050-2454/adba7>.
42. Chachra, A.; Kumar, A.; Ram, M.; Triantafyllou, I.S. Statistical fuzzy reliability assessment of a blended system. *Axioms* **2023**, *12*, 419. <https://doi.org/10.3390/axioms12050419>.

43. Akhtar, I.; Kirmani, S. An application of fuzzy fault tree analysis for reliability evaluation of wind energy system. *IETE J. Res.* **2022**, *68*, 4265–4278. <https://doi.org/10.1080/03772063.2020.1791741>.
44. Liu, C.; Zhou, C.; Wang, H.; Liu, S.; Cui, J.; Zhao, W.; Liu, S.; Tan, L.; Xiao, W.; Chen, Y. Reliability analysis of subsea pipeline system based on fuzzy polymorphic bayesian network. *Sci. Rep.* **2025**, *15*, 11523. <https://doi.org/10.1038/s41598-025-92588-3>.
45. Leoni, L.; De Carlo, F. Integration of fuzzy reliability analysis and consequence simulation to conduct risk assessment. *J. Loss Prev. Process Ind.* **2023**, *83*, 105081. <https://doi.org/10.1016/j.jlp.2023.105081>.
46. Alabool, H.M.; Alarabiat, D.; Abualigah, L.; Heidari, A.A. Harris hawks optimization: A comprehensive review of recent variants and applications. *Neural Comput. Appl.* **2021**, *33*, 8939–8980. <https://doi.org/10.1007/s00521-021-05720-5>.
47. 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. <https://doi.org/10.1016/j.future.2019.02.028>.
48. Ouyang, C.; Liao, C.; Zhu, D.; Zheng, Y.; Zhou, C.; Li, T. Integrated improved Harris hawks optimization for global and engineering optimization. *Sci. Rep.* **2024**, *14*, 7445. <https://doi.org/10.1038/s41598-024-58029-3>.
49. Akl, D.T.; Saafan, M.M.; Haikal, A.Y.; El-Gendy, E.M. IHHO: An improved Harris Hawks optimization algorithm for solving engineering problems. *Neural Comput. Appl.* **2024**, *36*, 12185–12298. <https://doi.org/10.1007/s00521-024-09603-3>.
50. Elgamal, Z.M.; Yasin, N.B.M.; Tubishat, M.; Alswaitti, M.; Mirjalili, S. An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. *IEEE Access* **2020**, *8*, 186638–186652. <https://doi.org/10.1109/ACCESS.2020.3029728>.
51. Abdel-Basset, M.; Ding, W.; El-Shahat, D. A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection. *Artif. Intell. Rev.* **2021**, *54*, 593. <https://doi.org/10.1007/s10462-020-09860-3>.
52. Al-Adwan, S.; Abdullah, S.; Alweshah, M.; Kassaymeh, S.; Al-Qaisi, A.; Takruri, M.; Ratan Singh, W.K. A multi-strategy framework for enhancing Harris hawks optimization for global optimization problems. *Sci. Rep.* **2026**, *16*, 10614. <https://doi.org/10.1038/s41598-026-45741-5>.
53. 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. <https://doi.org/10.1016/j.cie.2021.107408>.
54. Lang, Y.; Gao, Y. Dream Optimization Algorithm (DOA): A novel metaheuristic optimization algorithm inspired by human dreams and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **2025**, *436*, 117718. <https://doi.org/10.1016/j.cma.2024.117718>.
55. Wang, J.; Gao, Y.; Qin, L.; Li, Y. Logistic-gauss circle optimizer: Theory and applications. *Appl. Math. Model.* **2025**, *143*, 116052. <https://doi.org/10.1016/j.apm.2025.116052>.

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.