

Article

Chemical Safety Inspection Path Optimization Problems Using Improved Multi-Objective Discrete Growth Optimization Algorithm

Shanshan Luo ¹, Qiang Liu ^{1,*}, Xiwang Guo ¹, Mingqiang Yin ¹, Zhiwu Li ² and Xianming Lang ¹¹ School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China² Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China

* Correspondence: neuliuqiang@163.com

Abstract: Robot path planning plays a critical role in enhancing the efficiency and accuracy of inspections and ultimately contributes to the safety of chemical production. In particular, the performance of robot inspection is influenced by amount of gas leakage and path length. In this study, a chemical safety inspection path multi-objective optimization model is constructed that considers minimizing path length and maximizing gas leakage detection sensitivity. To address this model, an improved multi-objective discrete growth optimization algorithm is proposed, which adopts discrete operators to meet the solution requirements of the discrete model and an adaptive leader selection strategy to further improve the algorithm solution performance. Finally, numerical computations and simulation experiments are conducted to validate the feasibility of the proposed method for inspection path planning. The results demonstrate that the method outperforms multi-objective discrete particle swarm optimization, non-dominated sorting genetic algorithm-II, and multi-objective gray wolf optimization in terms of solution convergence and distribution uniformity. Moreover, it can provide better non-dominated solutions in simulation experiments and provide multiple references for practical applications.

Keywords: chemical safety; inspection path; multi-objective optimization; robot path planning; discrete operator; adaptive leader selection strategy



Received: 23 March 2025

Revised: 14 April 2025

Accepted: 30 April 2025

Published: 9 May 2025

Citation: Luo, S.; Liu, Q.; Guo, X.; Yin, M.; Li, Z.; Lang, X. Chemical Safety Inspection Path Optimization Problems Using Improved Multi-Objective Discrete Growth Optimization Algorithm. *Processes* **2025**, *13*, 1445. <https://doi.org/10.3390/pr13051445>

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

1. Introduction

Chemical use poses risks of casualties and property losses in the event of accidents [1]. Therefore, ensuring chemical safety has become a key issue. In particular, chemical safety inspections are an effective means of mitigating damage and preventing accidents. It can identify potential hazards and detect initial accidents. There are two main methods of chemical safety inspection: traditional inspection methods that rely on human labor and intelligent inspection methods that use robots. However, human inspectors are prone to errors, especially when fatigued. This compromises inspection accuracy and efficiency. The emergence of intelligent robot inspection has transformed chemical safety inspections [2]. It overcomes the limitations of manual methods. By replacing manual labor with robots, inspection tasks are completed more efficiently, particularly with the implementation of autonomous patrol capabilities.

When the robot performs automatic inspection, the inspection standard significantly impacts its inspection efficiency and its ability to prevent safety accidents. Li et al. [3] proposed a robotic inspection method based on fire risk levels. This significantly enhances inspection efficiency compared to manual approaches. However, this method uses fire level as the inspection criterion, focusing solely on fire accidents and offering limited effectiveness in preventing chemical safety accidents. Additionally, gas leakage is a common factor in fire accidents and can lead to catastrophic events such as explosions and poisoning [4]. Considering the cause of the occurrence of accidents, using leaking gas as a criterion for robotic inspection allows for timely monitoring of accident conditions and trends, thereby more effectively predicting and preventing hazardous accidents. Therefore, this study explores the factor of gas leakage and introduces detection sensitivity as an additional objective alongside path length. It can further enhance the efficiency of chemical safety inspections.

Additionally, path planning plays a crucial role in the automated inspection of chemical safety robots. It significantly impacts the quality, efficiency, and energy consumption of inspections. A robust path selection algorithm is fundamental and provides essential support for robot inspection. Over the past decades, various algorithms have been used to minimize the path length in path planning, including graph-based algorithms [5,6], sampling-based algorithms [7], and heuristic algorithms [8]. Furthermore, the emergence of intelligent algorithms, such as genetic algorithm-based approaches [9], ant colony optimization-based techniques [10], and particle swarm optimization-based strategies [11], has contributed to this objective. With further research, the objective of considering only the single shortest path in some cases can no longer meet the requirements. Liu et al. proposed a Kriging model and non-dominated sorting genetic algorithm-II based on the Kriging model for solving the problem by considering pipeline resonance in aero-engine pipeline routing [12]. Xie et al. proposed an improved ant colony optimization algorithm for solving the multi-objective path planning problem of inspection in radioactive environments by considering the radiation dose factor in inspection in the nuclear industry [13]. Zheng et al. proposed a Gaussian adaptive strategy-based multi-objective evolutionary optimization to balance path uniformity, slope, and undulation in uneven terrain planning [14]. Huang et al. adopted the improved multi-objective particle swarm method to minimize the total transportation cost and total transportation time as the objective function [15]. In addition, multi-objective algorithms, such as multi-objective scrambled frog jump algorithm [16], multi-objective adversarial robust convolutional neural network [17], multi-objective discrete artificial bee colony algorithm [18], and multi-objective evolutionary algorithm [19], also provide references for multi-objective path planning problems.

Path planning algorithms have been the subject of extensive research and development, yielding complex techniques for application in a variety of problems. Despite these advancements, their practical utilization in the inspection tasks undertaken by chemical safety robots remains somewhat limited. For example, how the population is encoded and decoded when the algorithm solves the problem, and how the algorithm is discretized. This represents an open field of research. This study considers maximizing gas leakage detection sensitivity and minimizing path length as objectives. Then, it establishes a multi-objective optimization model for chemical safety inspection path planning. To tackle this multi-objective optimization problem, an improved multi-objective discrete growth optimization algorithm is proposed based on growth optimization. Growth optimization

draws from individuals' reflective and learning mechanisms as they grow up in society [20]. This approach offers several advantages, including few parameters, fast convergence, and high solution accuracy. Its application has been successful in several fields, such as solar photovoltaic cells' parameter identification and modules [21] and intrusion detection systems [22]. Given its strong ability to escape local optima, growth optimization is deemed suitable for solving multi-objective optimization problems. Growth optimization is extended and improved to address the chemical safety path optimization problem. This has never been seen in the existing literature. In addition, selecting a leader by a hierarchical evolutionary state detection mechanism [23] effectively improves the problem-solving ability of the algorithm. The means of hyper-cone domain and aggregation [24] also contribute to this improvement. These examples illustrate that the quality of the algorithmic solution can be further enhanced by using a good leader selection strategy. Therefore, an adaptive leader selection strategy based on Q-learning is proposed and applied in this work.

The contributions of this work are given below: (i) Given the impact of gas leakage diffusion on the accuracy and efficiency of chemical safety robot inspection, detection sensitivity is formulated by introducing the Gaussian model of gas diffusion. Subsequently, to minimize the inspection path length while maximizing detection sensitivity, a multi-objective optimization model for the chemical safety inspection path planning problem is established. (ii) A multi-objective algorithm is proposed based on standard growth optimization, incorporating discrete operators to discretize the standard algorithm and subsequently using the concept of non-dominated solutions to extend it to the multi-objective optimization domain. The Latin Hypercube method is employed for population initialization, and an adaptive leader selection strategy based on Q-learning is proposed to further enhance the algorithm's solution performance. (iii) Numerical computations involving varying numbers of risk points and an inspection simulation are performed on the established model. The results demonstrate that the improved multi-objective discrete growth optimization algorithm (IMDGO) is feasible and surpasses other algorithms in effectively solving the formulated chemical safety inspection path planning problem.

This work consists of five sections. Section 2 describes the problem under the study and establishes a model of the problem. Section 3 describes the proposed methodology in detail. Then, the method's feasibility is demonstrated through numerical computations and the inspection simulation is performed on the established model in Section 4. Finally, Section 5 draws a conclusion and outlines future work.

2. Problem Description

2.1. Problem Statement

With the advancement of artificial intelligence and robotic technology, robot inspections are increasingly replacing labor inspections in industries. Currently, robot inspection relies on path planning to accomplish its inspection tasks. However, path planning methods in chemical safety primarily focus on identifying collision-free routes with the shortest path length. It overlooks the impact of gas leakage diffusion into the environment on inspection efficiency and accuracy. Consequently, incorporating the detection of gas leakage to find an optimal path can enhance the efficiency and accuracy of robotic inspections in the chemical industry.

This work uses the detection sensitivity to define the detection of gas leakage. The detection sensitivity of gas leakage depends on the distance between the risk point (i.e., the point of gas leakage) and the robot. It also relies on the effective detection duration of the sensor. Assuming the robot moves at a constant speed, the effective detection distance instead of the effective detection duration can be used. In this work, the road closest to the risk point is considered as the effective detection distance. Additionally, detection

sensitivity is defined by the Gaussian model of gas diffusion [25], as shown in Equation (1). During the inspection process, achieving high detection sensitivity within a certain path distance can improve inspection accuracy and effectively prevent chemical safety accidents. Therefore, this work aims to maximize detection sensitivity and minimize the path length to enhance chemical safety prevention.

$$C_{ijp} = \int_{L_{ijp}} \left(u \frac{Q}{d_{ijpj'}^\alpha} + w \right) ds \quad (1)$$

where C_{ijp} denotes the detection sensitivity of the path p from point i to point j , L_{ijp} represents the road closest to risk point j' , u stands for the parameter of gas diffusion in the current environment, Q indicates the equivalent signal strength of the leakage source, $d_{ijpj'}$ implies the distance from robot to risk point j' , α means the attenuation factor, and w is the random noise of gas in the diffusion process. And, the formula assumes that the leakage process of gas is a constant-state gas diffusion process, where the concentration of the leakage source is invariable and the differences in local temperature due to pressure changes before and after gas leakage are ignored.

In addition, in chemical inspection tasks, the robot always starts from a certain position (e.g., the robot standby or stored warehouse), then detects all the risk points, and finally returns to that point. During this process, all risk points should be visited once, and the nodes on a certain path between two risk points are allowed to be visited repeatedly. Therefore, the problem under study in this work can be considered to be a variant of the Traveling Salesman Problem (TSP).

2.2. Simulation Model

In this work, an inspection simulation model is used to simulate an actual chemical factory as shown in Figure 1. The model is built by reading the coordinate information stored in the document and connecting these coordinates according to the actual road distribution of the factory. In particular, the gray area is used to represent the work area of a chemical factory, which is an inaccessible area for robots and contains pipelines or reactors that are prone to gas leakage. The red lines represent the roads that connect the factory's road nodes, and the robots can walk on these roads. Triangles are used to indicate the risk points where leaks occur. Since the risk point area is an area that the robot cannot reach, the point closest to the risk point on the road is selected and defined as an adjacent point. These adjacent points are then replaced with risk points to visit, and the robot can complete the inspection task by visiting these adjacent points. In addition, as the gas leaks into the air, the concentration of the contained gas varies at different distances from the risk point. From Equation (1), the closer to the risk point, the higher the gas concentration and the higher the gas sensitivity, and vice versa. So, the edges containing adjacent points are distinguished by different colors in Figure 1 compared to other edges. This distinction indicates that these edges can be used as effective detection edges for the robot. In addition, the depth of the color of the edge where the adjacent points are located indicates the concentration of the leaked gas after diffusion. In other words, a darker color indicates a higher concentration of leaking gas at these locations.

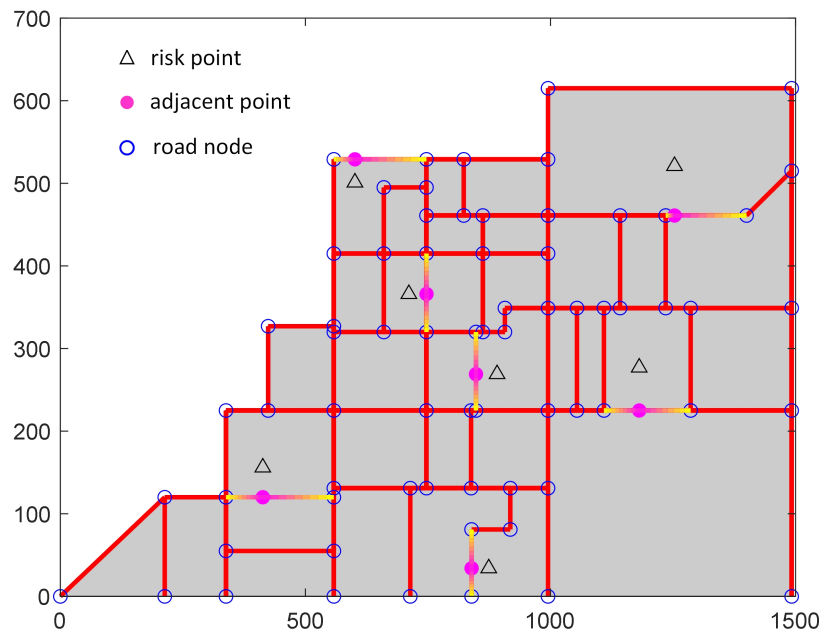


Figure 1. Inspection simulation model.

2.3. Decision Variables and Objective Functions

For clarity of presentation, all notations are as follows:

A Set of adjacent points and start point. In particular, the start point is defined as 1, adjacent points are defined 2 to $N + 1$, and N denotes the total number of adjacent points.

S Subset of A .

P_{ij} Set of path between two different adjacent points or between the start point and the adjacent point.

i, j Index of adjacent points and start point. They all belong to set A .

p Index of the path between two different adjacent points or the start point to adjacent point, which belongs to set P_{ij}

D_{ijp} Distance of the path p from point i to point j .

C_{ijp} Detection sensitivity of the path p from point i to point j .

The decision variable is as follows:

$$x_{ijp} = \begin{cases} 1, & \text{if the path } p \text{ between point } i \text{ and } j \text{ is selected} \\ 0, & \text{otherwise} \end{cases}$$

With the above description and definition, a chemical safety inspection path optimization problem can be specified by establishing a mathematical model. The model is presented as follows:

$$\min f_1 = \sum_{i \in A} \sum_{j \in A} \sum_{p \in P_{ij}} x_{ijp} D_{ijp} \quad (2)$$

$$\max f_2 = \sum_{i \in A} \sum_{j \in A/\{1\}} \sum_{p \in P_{ij}} x_{ijp} C_{ijp} \quad (3)$$

$$\sum_{j \in A/\{i\}} \sum_{p \in P_{ij}} x_{ijp} = 1, \forall i \in A \quad (4)$$

$$\sum_{j \in A/\{i\}} \sum_{p \in P_{ij}} x_{jip} = 1, \forall i \in A \quad (5)$$

$$\sum_{i,j \in S} \sum_{p \in P_{ij}} x_{ijp} \leq |S| - 1 \quad (6)$$

$$x_{ijp} \in \{0, 1\}, \forall i, j \in A, \forall p \in P_{ij} \quad (7)$$

The objective function (2) denotes the minimum path length of robot inspection, in which D_{ijp} is the distance of path p between point i and j . The objective function (3) presents the maximum detection sensitivity by the sensor carried by the robot. Constraints (4) and (5) indicate that the robot has to go to the adjacent point to complete the detection of the risk point. Constraint (6) indicates that no subtour can occur. Constraint (7) is a decision variable constraint for 0–1 integers.

3. Proposed Algorithm

The chemical safety inspection path planning method is divided into two main stages. The first stage is the modeling stage. In this stage, the inspection environment model is established, the risk points are set, and then the adjacent points are calculated. The adjacent matrix is computed using A* as a parametric input to the multi-objective algorithm for solving the problem. The second stage is the algorithm solution stage that solves the model. The flowchart of the robot inspection path planning method is shown in Figure 2. The modeling is described in Section 2, and the solution algorithm improvement and process are described in detail in this section.

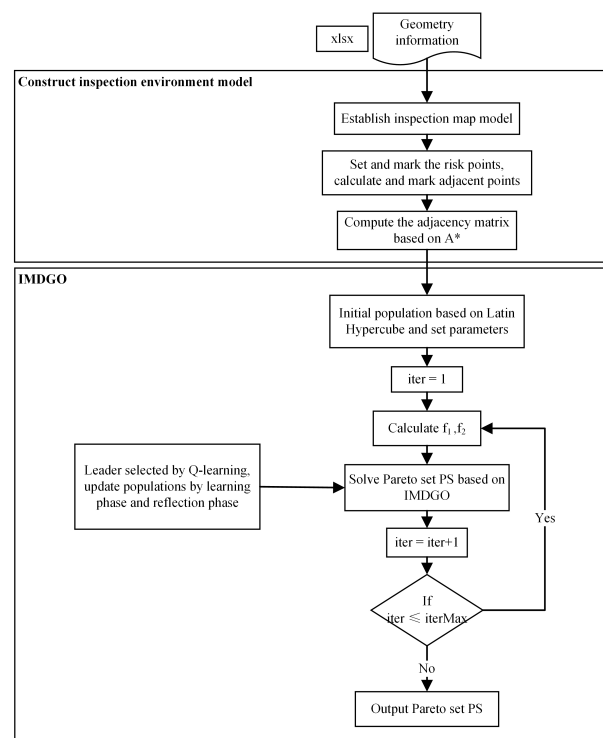


Figure 2. A flowchart of the robot inspection path planning method.

3.1. Population Initialization

3.1.1. Initialization Method

The Latin Hypercube method [26] is a stratified sampling method that creates uniform sampling points in the solution space and avoids the aggregation phenomenon associated with random sampling. The diversity of the population is enhanced and premature convergence of the algorithm to a local optimum is effectively prevented by the method. As an example, Figure 3 compares simple random sampling with Latin Hypercube sampling in two-dimensional coordinates.

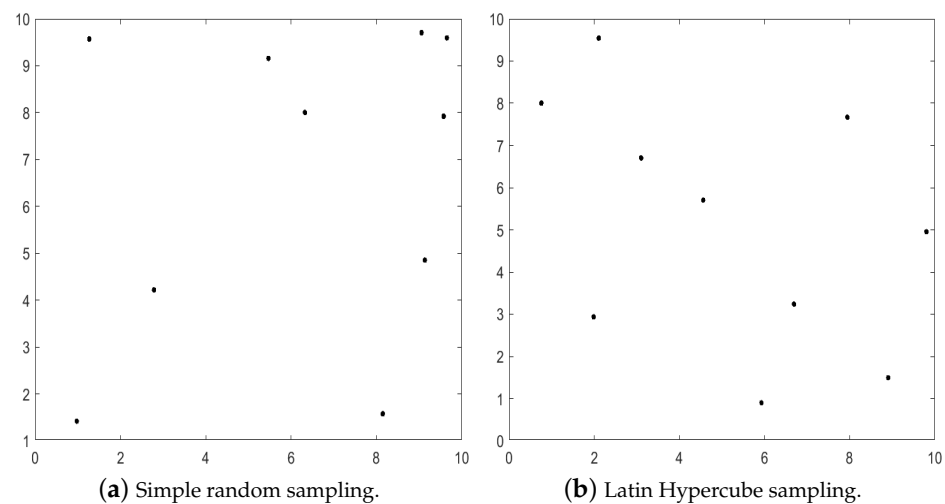


Figure 3. Sample distribution chart.

3.1.2. Encoding and Decoding

The problem studied in this work belongs to a variant of the TSP, so sequential encoding should be used for encoding. Let X be an m -dimensional vector, where each dimension's data represent the index of an adjacent point, and also both the endpoint of the previous path and the starting point of the next path.

$$X = (x_1, x_2, \dots, x_i, \dots, x_N), 1 \leq i \leq N, 2 \leq x_i \leq N + 1 \quad (8)$$

The robot always starts from a certain position (e.g., the robot standby or stored warehouse) and then detects all the risk points to return to that position to complete an inspection task in the chemical industry. Therefore, in the process of decoding, the index of the starting position is assigned as 1, and then that number is placed in the first and last dimension of X . The whole sequence of visits becomes a cyclic singly linked list, indicating that the robot starts from the starting point, passes through and detects all the adjacent points, and finally returns to the starting point, completing an inspection task.

3.2. Leader Selection Strategy

In multi-objective algorithms, as the number of iterations increases, the population may fall into a local optimum, resulting in a lack of diversity in the updated population. A suitable leader selection strategy with the ability to improve the convergence of the population to the true Pareto Frontier can effectively solve this problem. In general, the leader of the population is chosen from the Pareto set (PS) using roulette wheel selection or grid selection. However, in this work, leaders are selected using an adaptive leader selection strategy based on Q-learning, which aims to balance the convergence and diversity of the population, considering the advantages of the Q-learning algorithm such as its easy

implementation and model-free learning capability. The detailed process of Q-learning is described by Guo et al. [27].

First, non-dominated solutions generated in the external archive are used as candidates for the leaders. Second, the population is divided into several groups, with the number of individuals in each group not exceeding the number of non-dominated solutions. Once several groups have been formed, the adjacent matrix is determined based on the Manhattan distance between the individuals in each group and the non-dominated solutions. Finally, the minimum sum of each group of distances is taken as the objective and solved using Q-learning. Find the correspondence between the individual and the non-dominated solution in the obtained final solution and use the non-dominated solution as the leader of the corresponding individual. The chart of the adaptive leader selection strategy is shown in Figure 4.

The advantages of this strategy are as follows: in early iterations, the number of non-dominated solutions is less than the number of individuals in the population, and the nearest non-dominated solution is selected as the leader for each group, which ensures that limited resources are utilized to the maximum, and increases the exploration ability of the algorithm. In later iterations, the number of non-dominated solutions is not less than the number of individuals in the population, the same individual can choose different leaders, and different individuals can choose the same leader, which ensures the exploitation ability of the algorithm and adds a certain exploration ability. The process of leader selection in different iteration periods is shown in Figure 5. The numbers in the brackets in the figure represent the number of corresponding solutions. Overall, the proposed adaptive leader selection strategy effectively balances the convergence and diversity of the population.

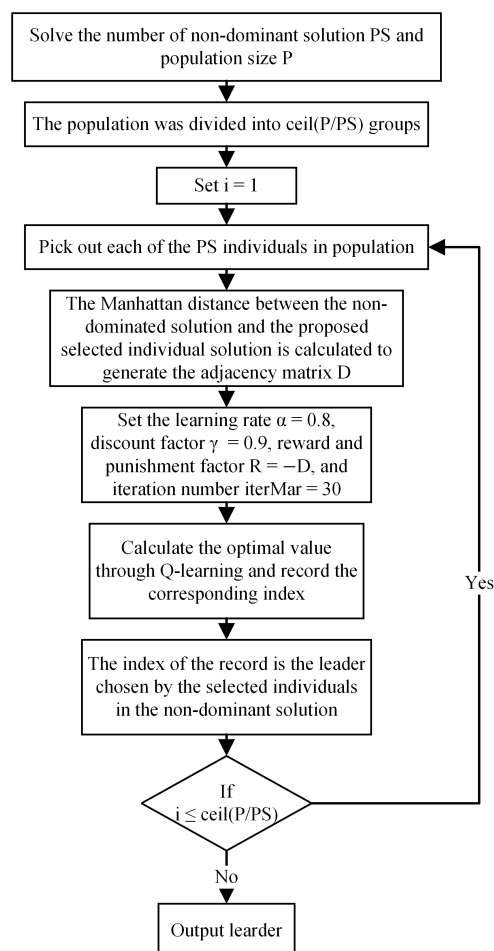


Figure 4. Adaptive leader selection strategy.

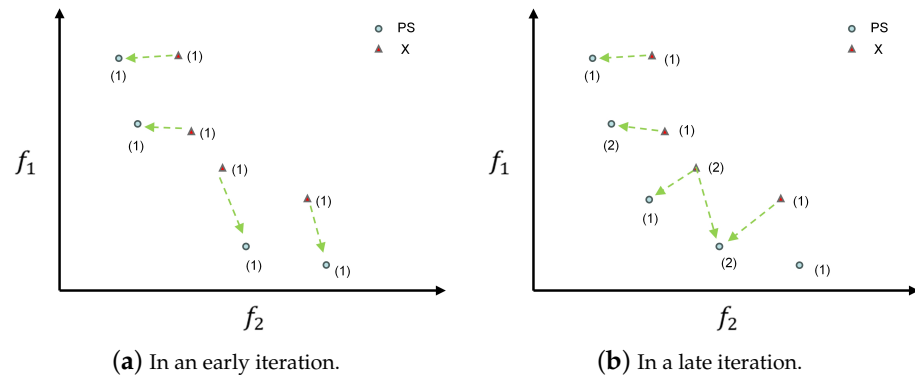


Figure 5. Leader selection sample chart.

3.3. Algorithm Discretization

According to Section 2, it can be learned that the problem falls under the category of a variant of TSP. Since growth optimization is designed to solve continuous optimization problems, it needs to be discretized to tackle this discrete problem. Consequently, some discrete operators [28] have been devised as follows.

3.3.1. Swap Index Pair (S^{IP})

The swap index pair S^{IP} is formed between two individuals and calculated by Equation (9). Its solution process is shown in Figure 6. The specific steps are as follows:

Step 1: In the selected population, individual S_1 and individual S_2 are selected by the rules of the growth optimization.

Step 2: Find the data point in individual S_2 that is equal to the dimension data point of individual S_1 , and then record the index of that data point.

Step 3: The odd positions of S^{IP} are filled from 1 to the dimension size of individual S_1 , and the even positions of S^{IP} are filled sequentially by the index recorded in Step 2.

$$S^{IP} = S_1 \sim S_2 = (2, 1, 3, 4) \sim (3, 4, 2, 1) = (1, 3, 2, 4, 3, 1, 4, 2) \quad (9)$$

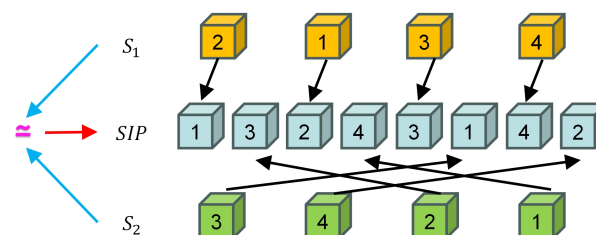


Figure 6. An example of swap index pair process.

3.3.2. Swap Section (S^{ST})

The swap section S^{ST} indicates a part of S^{IP} , for which the calculation is shown in Equation (10) and the solution process is shown in Figure 7.

$$S^{ST} = r \times S^{IP} = 2 \times (1, 3, 2, 4, 3, 1, 4, 2) = (1, 3, 2, 4) \quad (10)$$

We can see that S^{ST} is generated by intercepting the first r pairs of data from S^{IP} . A learning factor LF_k in the growth optimization could be discretized to denote r , which is shown in Equation (11).

$$LF_k = \text{round}\left(\frac{\text{num}_k}{\sum_{k=1}^4 \text{num}_k} \times N\right) \quad (11)$$

where *round* denotes the rounding operation to ensure that LF_k is an integer, and num_k is a counter that records the number of index differences when calculating the gap between individuals by Equation (9).

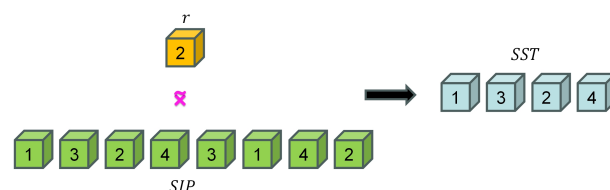


Figure 7. An example of swap section process.

3.3.3. Swap Sequence (S^S)

S^S is made up of one or more S^{ST} s and its solution process is shown in Figure 8. Here, the order of the S^{ST} in S^S is important and cannot be changed.

$$S^S = (S_1^{ST}, S_2^{ST}, \dots, S_k^{ST}) \quad (12)$$

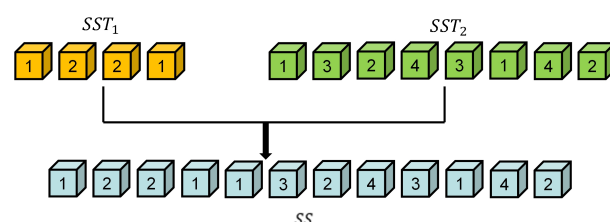


Figure 8. An example of swap sequence process.

3.3.4. Swap Operation (S^O)

S^O denotes an index of positions in the individual that might be swapped for a new individual from an individual by using S^S . The probabilistic choice of SF is decisive for the computation of the new individual S' , whose discretization is shown in Equation (13). Firstly, a pair of index loops are selected from S^S for individual updates according to Equation (10), where $r = 1$ and $S^{IP} = S^S$. Continuous probabilistic selection for updating occurs in a loop until there are no index pairs available for updating. If the uniformly distributed random number in the range $[0, 1]$ is less than SF , the individual is updated based on the index pair, whose update formula is shown in Equation (14) and Figure 9, and the update process is to swap the data point in the index pair corresponding to the old individual S . If not, update without the index pair.

$$SF = \begin{cases} \frac{num_2}{N} & \text{if } num_2 \neq 0 \\ rand & \text{else} \end{cases} \quad (13)$$

$$S' = S \dot{+} S^{ST} = (1, 2, 3, 4) \dot{+} (2, 3) = (1, 3, 2, 4) \quad (14)$$

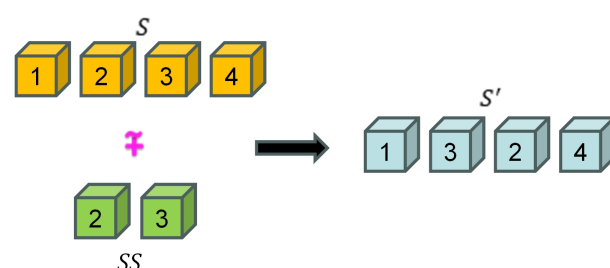


Figure 9. An example of swap operation process.

3.4. Algorithm Implementation

The IMDGO process involves the following three steps: (1) initializing a society population, (2) selecting the leader and other individuals, and (3) updating the population, as implemented in Algorithm 1.

Algorithm 1 Improved multi-objective growth optimization algorithm

Require: adjacent matrix, maximum number of iterations T , population size, P_2

Ensure: PS = the set of pareto

```

1: Initialize the society population
2: Calculate the fitness  $f_1$  and  $f_2$  of each individual
3: Find all non-dominated solutions and add them to  $PS$ 
4: while  $t < T$  do
5:   Select leader  $X_{best}$  of each individual based on Figure 4
6:   for  $k = 1$  to  $P$  do
7:     Select  $X_{better}$  in historical non-nominated solutions
8:      $X_{worse}$  is the  $i$ -th individual
9:     Find two random individuals that are different from  $X_{worse}$ :  $X_{L1}$  and  $X_{L2}$ 
10:    Calculate  $Gap$  based on Equation (9)
11:    Calculate  $LF$  based on Equation (11)
12:    Calculate  $KA$  based on Equations (10) and (12)
13:    Calculate  $SF$  based on Equation (13)
14:    if  $rand < SF$  then
15:      Calculate the new individual based on Equation (14)
16:    end if
17:    Calculate the fitness  $f_1$  and  $f_2$  of the new individual
18:    Determine whether the new individual dominates the old one
19:    if the new individual dominates the old individual or  $rand < P_2$  then
20:      Update the individual: the new individual
21:    else
22:      Calculate  $AF = 0.2 + 0.8 * (1 - t/T)$ 
23:      if  $rand < AF$  then
24:        Update the individual based on random sequences and Equation (14)
25:      else
26:        Update the individual based on random individual and Equation (14)
27:      end if
28:    end if
29:    Calculate the fitness  $f_1$  and  $f_2$  of the individual
30:  end for
31:  Find all non-dominated solutions and add them to  $PS$ 
32:   $t = t + 1$ 
33: end while
34: return  $PS$ 

```

4. Numerical Experiments

To demonstrate the effectiveness of the proposed algorithm, some numerical computations and an inspection simulation on a model of the inspection environment given in Figure 1 are performed. The computations are executed on a PC (AMD Ryzen 5 5600U with Radeon Graphics@2.30 GHz, 16.0 GB RAM). The software uses MATLAB R2022b. The experimental cases with different numbers of risk points in the chemical safety inspection

environment model are solved. A case is selected as an inspection simulation to verify the feasibility of the proposed algorithm.

4.1. Numerical Computations

In this work, different numbers of risk points are selected for testing in the established model. The C-metric (coverage metric between two solution sets) [29] and the S-metric (spacing metric) [30] are used to evaluate the convergence of the solution and the uniformity of the solution distribution, respectively.

To compare the performance of IMDGO, multi-objective discrete particle swarm optimization (MDPSO) [31], non-dominated sorting genetic algorithm-II (NSGA-II) [32], and multi-objective gray wolf optimization (MOGWO) [33], the maximum number of iterations is set to the same value of 200. The parameter P_2 is set to 0.2 in IMDGO, the inertia weight W is linearly reduced from 0.9 to 0.2, and learning rates are all set to 1 in MDPSO. The crossover probability and the mutation probability are, respectively, set to 0.9 and 0.1 in NSGA-II; α , β , and γ are, respectively, set to 0.1, 4, and 2 in MOGWO. In this work, the size of population P is set to 100, 120, or 150, each algorithm is run independently ten times for comparison, and the data obtained from each run are recorded. Tables 1 and 2 give the comparisons of the C-metric and S-metric of the sets solved by IMDGO and the other three algorithms, respectively.

Table 1 presents the C-metric experimental results of the four algorithms. The numbers of risk points are set to 10, 15, and 20, individually. In the header row of the table, P_1 , P_2 , P_3 , and P_4 denote the set of non-dominated solutions obtained by IMDGO, MDPSO, NSGA-II, and MOGWO computations, respectively. From the results of these experiments, it can be known that IMDGO is better than the other three algorithms in terms of the best, mean, and worst C-metric values, which indicates that the convergence of IMDGO is better than that of the other three algorithms. In particular, $C(P_1, P_2)$, $C(P_1, P_3)$, and $C(P_1, P_4)$ reached 1 for both 15 and 20 risk points in terms of the best C-metric values, which indicates that the solution sought by the comparison algorithm is only a fraction of the solution sought by the proposed algorithm. The superiority of the proposed algorithm in terms of solution quality is more clearly shown and becomes more obvious as the number of risk points increases. Therefore, it is concluded that IMDGO is superior to those of its peers in finding optimal solutions to the problem presented in this work.

Table 2 presents the S-metric experimental results of the four algorithms. The numbers of risk points are set to 10, 15, and 20, individually. For each case with different numbers of risk points, the mean and worst S-metric values calculated by IMDGO are smaller than those of its peers. While its best S-metric values are slightly worse than those of NSGA-II, they are still smaller than those of the other two algorithms. The results show that the non-dominant solution set obtained by the IMDGO algorithm has a good distribution. And, the solution distribution obtained by the IMDGO algorithm is more stable from its mean value, so it can be inferred that IMDGO algorithm is better than similar algorithms in most cases. However, it can be inferred from the fact that its worst value distribution solution is inferior to NSGA-II that it does not always outperform its peers. In addition, the S-metric value tends to become smaller as the number of risk points increases, which indicates that the solution of the proposed algorithm is more uniformly distributed as the number of risk points increases.

Based on the experimental results presented in Tables 1 and 2, it can be inferred that although IMDGO may not outperform other algorithms in all scenarios, it provides competitive solutions in most cases. In addition, the performance of IMDGO gradually improves as the number of risk points increases, indicating that the algorithm has a high degree of scalability.

Table 1. Comparison of four algorithms via C-metric.

Case	P	$C(P_1, P_2)$			$C(P_2, P_1)$			$C(P_1, P_3)$		
		Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
10	100	1	0.72917	0.3333	0.5000	0.17818	0	1	0.57695	0.1000
	120	1	0.81970	0.3636	0.3333	0.05492	0	0.8	0.52434	0.1111
	150	0.9	0.74432	0.6000	0.3333	0.18102	0	0.8333	0.44265	0.1250
15	100	1	0.95714	0.5714	0.1667	0.01667	0	1	0.55071	0.1000
	120	1	0.86515	0.3333	0.1667	0.01667	0	1	0.74210	0.5000
	150	1	0.96071	0.7500	0.0909	0.00909	0	1	0.69476	0.2500
20	100	1	0.95500	0.7500	0	0	0	1	0.65790	0.2500
	120	1	0.94205	0.7273	0	0	0	1	0.71879	0.0909
	150	1	0.95953	0.6667	0	0	0	1	0.60493	0.2000
Case	P	$C(P_3, P_1)$			$C(P_1, P_4)$			$C(P_4, P_1)$		
		Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
10	100	0.6000	0.09333	0	1	0.61638	0.3000	0.7500	0.31548	0
	120	0.5000	0.10833	0	0.8571	0.45857	0.1818	0.7143	0.45286	0
	150	0.5000	0.17333	0	0.8333	0.46781	0.1250	0.7500	0.36523	0
15	100	0.8000	0.14666	0	1	0.6975	0.1250	0.8333	0.26444	0
	120	0.2000	0.03667	0	1	0.60146	0.2500	0.3571	0.15904	0
	150	0.3333	0.05333	0	1	0.76547	0.2857	0.6667	0.14667	0
20	100	0.6667	0.06667	0	1	0.67666	0.2857	0.5000	0.12732	0
	120	0.5000	0.05000	0	1	0.76505	0.3750	0.2000	0.03547	0
	150	0.6667	0.09167	0	1	0.90051	0.6364	0	0	0

Table 2. Comparison of four algorithms via S-metric.

Case	P	IMDGO			MDPSO			NSGA-II			MOGWO		
		Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst	Best	Mean	Worst
10	100	0.0454	0.10813	0.1983	0.0462	0.12620	0.2709	0	0.13778	0.4113	0.0533	0.12640	0.2562
	120	0.0304	0.10472	0.2191	0.0519	0.10875	0.2329	0	0.17185	0.4928	0.0467	0.14595	0.2670
	150	0.0117	0.09871	0.1626	0.0326	0.11701	0.2202	0.0272	0.10854	0.3	0.0667	0.10553	0.1636
15	100	0.0282	0.06374	0.1078	0.0545	0.15411	0.2880	0	0.12552	0.4524	0.0482	0.08728	0.1427
	120	0.0038	0.07577	0.1274	0.0480	0.11275	0.2070	0	0.07860	0.3353	0.0275	0.08407	0.1829
	150	0.0363	0.09315	0.1790	0.0525	0.11346	0.2094	0	0.12093	0.6115	0.0278	0.09655	0.2297
20	100	0.0094	0.07943	0.2316	0.0321	0.12548	0.2561	0	0.13098	0.3976	0.0225	0.15674	0.9260
	120	0.0284	0.07961	0.1369	0.0545	0.10837	0.1825	0	0.11885	0.4044	0.0345	0.08847	0.1687
	150	0.0377	0.08289	0.1655	0.0598	0.09533	0.2194	0	0.11439	0.3981	0.0395	0.09748	0.2134

4.2. Inspection Simulation

To intuitively demonstrate the performance and superiority of IMDGO, we have taken seven risk points as a sample to visualize a Pareto Front using four different algorithms. The non-dominated solutions and their corresponding inspection programs obtained by IMDGO are displayed. Moreover, the maximum number of iterations for IMDGO and the comparison algorithm is set to the same value of 200. The parameter P_2 is set to 0.2 in IMDGO, the inertia weight W is linearly reduced from 0.9 to 0.2, and learning rates are all set to 1 in MDPSO. The crossover probability and the mutation probability are, respectively, set to 0.9 and 0.1 in NSGA-II; α , β , and γ are, respectively, set to 0.1, 4, and 2 in MOGWO. In this work, the size of the population P is set to 100.

The Pareto Front is shown in Figure 10, which consists of the distribution of non-dominated solutions in objective space by different algorithms. The abscissa denotes path length, and the ordinate indicates detection sensitivity. It can be seen that the solutions obtained by IMDGO have a higher detection sensitivity at the same path length and a shorter path length at the same detection sensitivity. Therefore, IMDGO is superior to other algorithms in terms of the quality of solutions obtained for the problems presented in this work.

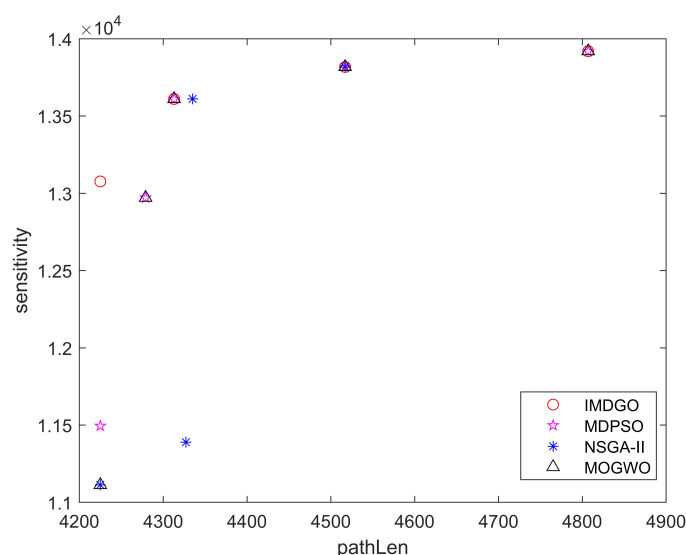
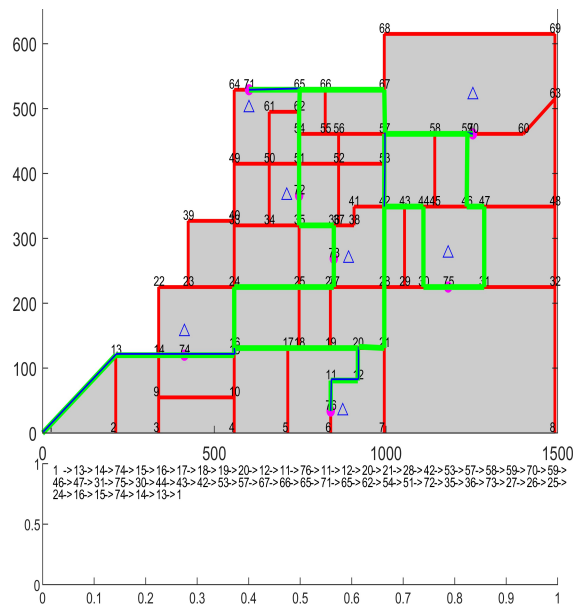
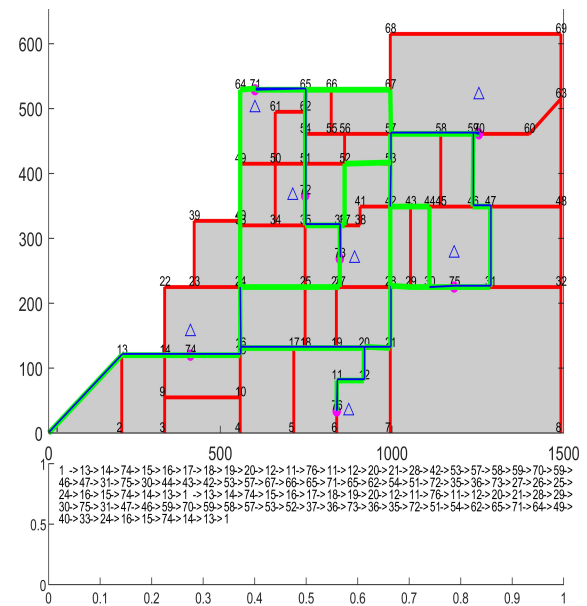
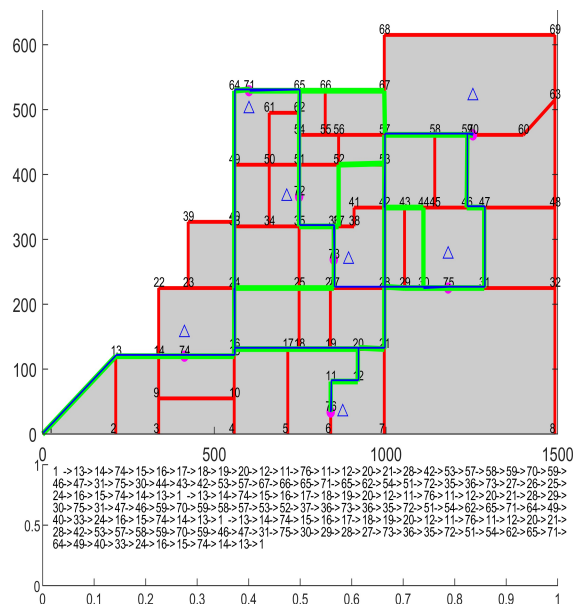
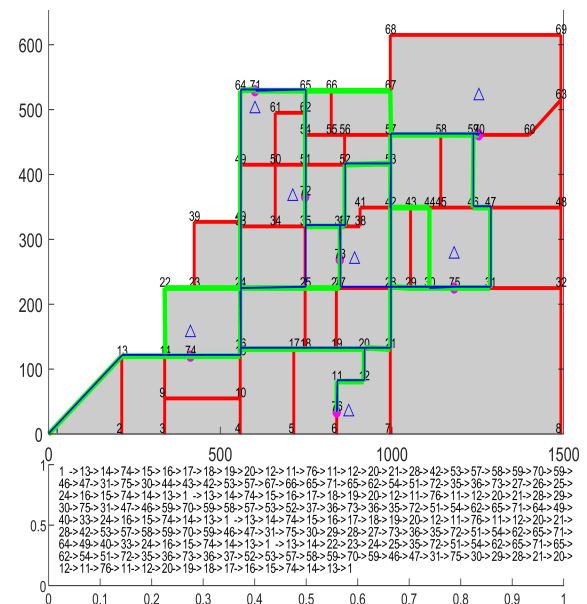


Figure 10. Non-dominated solutions in objective space from different algorithms.

Table 3 provides all solutions calculated by IMDGO in Figure 10, and Figure 11 displays the robot inspection programs for these solutions. In Figure 11, the green lines indicate roads where the robot will only pass once, and the blue lines show roads where the robot can go through repeatedly. The sequence of robot inspection nodes is presented by the numbers below each figure, which shows the process of robot inspection. A set of several non-dominated solutions with different trade-offs are obtained by the presented chemical safety inspection path optimization method in this work. The robot inspection process path length is smaller when the detection sensitivity is also smaller; at this time, the robot inspection time is shorter, but the detection sensitivity may not necessarily reach the actual need. When the path length is large, the detection sensitivity is also large. However, despite the large detection sensitivity, the robot inspection process requires more time. In practice, the goal will determine which inspection program is best to choose.

Table 3. Non-dominated solutions obtained by IMDGO.

PS Obtained	f_1	f_2
Non-dominated solution 1	4225	13076
Non-dominated solution 2	4313	13610
Non-dominated solution 3	4517	13817
Non-dominated solution 4	4807	13919

**(a)** Non-dominated solution 1.**(b)** Non-dominated solution 2.**(c)** Non-dominated solution 3.**(d)** Non-dominated solution 4.**Figure 11.** Robot inspection path of non-dominated solutions.

5. Conclusions

In this study, a mathematical model to solve the chemical safety inspection path optimization problem is established. A multi-objective optimization method for chemical safety inspection paths, focusing on both path length and detection sensitivity, is proposed based on IMDGO and further validated through an example of robot inspection in a chemical factory. The obtained solutions are satisfactory for each objective and can be tailored to real-world demands. It is worth noting that detection sensitivity is considered during the inspection process, which enhances the efficiency and accuracy of robotic inspection in chemical safety.

In particular, the standard growth optimizer has been improved. Considering the discreteness of the problem, some discrete operators are designed to discretize the standard growth optimizer. In the process of population initialization, the Latin Hypercube method is used to ensure the uniform distribution of individuals in the solution space, which improves the initial quality and diversity of the population. To balance the convergence and diversity of the population, an adaptive leader selection strategy based on Q-learning is proposed to further improve the performance of the algorithm.

There are numerous factors affecting chemical safety. Various path optimization methods have been considered for various problems, including but not limited to leaked gases, flammable and explosive materials, etc. In this work, in addition to considering the path length, the influence of leaked gas was also taken into account. However, any factor in the robot detection process is known and determined, and the lack of research on the impact of uncertainty on the detection process further enhances the efficiency and accuracy of the detection. The study in this paper was only conducted in a simulation model and further studies are needed in practical applications. Meanwhile, in response to the inspection requirements of large-scale factory areas, it is also necessary to study the multi-robot task allocation and path coordination strategies.

Author Contributions: S.L.: Data curation, Methodology, Writing—original draft. Q.L.: Conceptualization, Methodology, Supervision, Funding acquisition, Writing—review and editing. X.G.: Supervision, Writing—review and editing. M.Y.: Formal analysis, Funding acquisition. Z.L.: Supervision, Writing—review and editing. X.L.: Writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Basic Science Research Program for the Education Department of Liaoning Province under Grant No. JYTMS20231434, the Liaoning Provincial Science and Technology Innovation Project in the Field of Artificial Intelligence under Grant No. 2023JH26/10300013; and in part by the Liaoning Province Department of Education Project under Grant No. JYTQN2023345.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Bai, M.; Qi, M.; Shu, C.; Reniers, G.; Khan, F.; Chen, C.; Liu, Y. Why do major chemical accidents still happen in china: Analysis from a process safety management perspective. *Process Saf. Environ. Prot.* **2023**, *176*, 411–420. [[CrossRef](#)]
2. Vesentini, F.; Di, P.; Muradore, R. A Brownian–Markov stochastic model for cart-like wheeled mobile robots. *Eur. J. Control* **2023**, *70*, 100771. [[CrossRef](#)]
3. Li, Y.; Chen, S.; Bai, K.; Wang, H. Path planning of patrol robot based on improved discrete electrostatic discharge algorithm. *J. Intell. Fuzzy Syst.* **2022**, *42*, 5919–5930. [[CrossRef](#)]
4. Yu, J.; Zeng, Q.; Yu, Y.; Zhang, B.; Ding, H.; Ma, W.; Wu, S.; Li, Z. Safety barrier performance assessment based on multi-phase Markov model and EPN: Application to FPSO gas leakage. *Ocean Eng.* **2025**, *323*, 120597. [[CrossRef](#)]

5. Alina, B.; Antonio, C.; Antonino, G.; Jiafu, W.; Maria, F. Evaluating an application aware distributed dijkstra shortest path algorithm in hybrid cloud /edge environments. *IEEE Trans. Sustain. Comput.* **2022**, *7*, 289–298.
6. Liu, Q.; Wang, C. A graph-based pipe routing algorithm in aero-engine rotational space. *J. Intell. Manuf.* **2015**, *26*, 1077–1083. [[CrossRef](#)]
7. Dam, T.; Chalvatzaki, G.; Peters, J.; Pajarinen, J. Monte-carlo robot path planning. *IEEE Robot. Autom. Lett.* **2020**, *7*, 11213–11220. [[CrossRef](#)]
8. Rousseas, P.; Bechlioulis, C.; Kyriakopoulos, K. Harmonic-based optimal robot motion planning in constrained workspaces using reinforcement learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2005–2011. [[CrossRef](#)]
9. Chen, G.; Du, Y.; Xi, X.; Zhang, K.; Yang, J.; Xu, L.; Ren, C. Improved genetic algorithm based on bi-level co-evolution for coverage path planning in irregular region. *Sci. Rep.* **2025**, *15*, 10047. [[CrossRef](#)]
10. Fang, S.; Deng, Z.; Li, P.; Long, D. Improved strategy of ant colony optimization for path planning via stochastic pheromone updating and cyclic initialization. *J. Mech. Sci. Technol.* **2025**, *39*, 2051–2062. [[CrossRef](#)]
11. Shen, Y.; Zhang, W.; Zhang, Y.; Huang, N.; Zhang, Y.; Zhu, L. Distributed particle swarm optimization for the planning of time-optimal and interference-free five-axis sweep scanning path. *IEEE Trans. Ind. Inform.* **2022**, *18*, 8703–8713. [[CrossRef](#)]
12. Liu, Q.; Jiao, G. A pipe routing method considering vibration for aero-engine using kriging model and NSGA-II. *IEEE Access* **2018**, *6*, 6286–6292. [[CrossRef](#)]
13. Xie, X.; Tang, Z.; Cai, J. The multi-objective inspection path planning in radioactive environment based on an improved ant colony optimization algorithm. *Prog. Nucl. Energy* **2022**, *144*, 104076. [[CrossRef](#)]
14. Zheng, H.; Lu, Y.; Jie, J.; Hou, B.; Zhang, M.; Zhang, Y. Gaussian adaptive strategy based multi-objective evolutionary optimization for path planning on uneven terrains. *IEEE Robot. Autom. Lett.* **2024**, *9*, 539–546. [[CrossRef](#)]
15. Huang, C.; Sun, H.; Liu, C.; Zhang, X.; Gao, T.; Tian, J. Research on Container Multimodal Transportation Multi-Objective Path Optimization From Hinterlands to Shanghai Port. *IEEE Access* **2025**, *13*, 32794–32807. [[CrossRef](#)]
16. Guo, X.; Fan, C.; Zhou, M.; Liu, S.; Wang, J.; Qin, S.; Tang, Y. Human-robot collaborative disassembly line balancing problem with stochastic operation time and a solution via multi-objective shuffled frog leaping algorithm. *IEEE Trans. Autom. Sci. Eng.* **2023**, *21*, 4448–4459. [[CrossRef](#)]
17. Wei, H.; Zeng, G.; Lu, K.; Geng, G.; Weng, J. MoAR-CNN: Multi-Objective Adversarially Robust Convolutional Neural Network for SAR Image Classification. *IEEE Trans. Emerg. Top. Comput. Intell.* **2024**, *9*, 1–18. [[CrossRef](#)]
18. Li, X.; Ma, S. Multiobjective discrete artificial bee colony algorithm for multiobjective permutation flow shop scheduling problem with sequence dependent setup times. *IEEE Trans. Eng. Manag.* **2017**, *64*, 149–165. [[CrossRef](#)]
19. Xu, X.; Xie, C.; Ma, L.; Yang, L.; Zhang, T. mMulti-objective evolutionary algorithm with two balancing mechanisms for heterogeneous UAV swarm path planning. *Appl. Soft Comput.* **2025**, *173*, 112927. [[CrossRef](#)]
20. Zhang, Q.; Gao, H.; Zhan, Z.; Li, J.; Zhang, H. Growth optimizer: A powerful metaheuristic algorithm for solving continuous and discrete global optimization problems. *Knowl.-Based Syst.* **2023**, *261*, 110206. [[CrossRef](#)]
21. Ben Aribia, H.; El-Rifaie, A.; Tolba, M.; Shaheen, A.; Moustafa, G.; Elsayed, F.; Elshahed, M. Growth optimizer for parameter identification of solar photovoltaic cells and modules. *Sustainability* **2023**, *15*, 7896. [[CrossRef](#)]
22. Fatani, A.; Dahou, A.; Abd Elaziz, M.; Al-qaness, M.; Lu, S.; Alfadhli, S.; Alresheedi, S. Enhancing Intrusion Detection Systems for IoT and Cloud Environments Using a Growth Optimizer Algorithm and Conventional Neural Networks. *Sensors* **2023**, *23*, 4430. [[CrossRef](#)] [[PubMed](#)]
23. Han, H.; Zhang, L.; Yinga, A.; Qiao, J. Adaptive multiple selection strategy for multi-objective particle swarm optimization. *Inf. Sci.* **2023**, *624*, 235–251. [[CrossRef](#)]
24. Liu, J.; Liu, Y.; Han, H.; Zhang, X.; Shu, X.; Chen, F. A two-stage maintenance and multi-strategy selection for multi-objective particle swarm optimization. *Complex Intell. Syst.* **2023**, *9*, 7523–7548. [[CrossRef](#)]
25. Zhou, Y.; Ding, X.; Hu, Y. Research on method for locating ammonia leak source and development of related monitoring. *China Saf. Sci. J.* **2016**, *26*, 163–168.
26. Asef, P.; Denai, M.; Paulides, J.; Marques, B.; Laphorn, A. A novel multi-criteria local latin hypercube refinement system for commutation angle improvement in IPMSMs. *IEEE Trans. Ind. Appl.* **2022**, *59*, 1588–1602. [[CrossRef](#)]
27. Guo, X.; Bi, Z.; Wang, J.; Qin, S. Reinforcement learning for disassembly system optimization problems: A survey. *Int. J. Netw. Dyn. Intell.* **2023**, *2*, 1–14. [[CrossRef](#)]
28. Wang, K.; Huang, L.; Zhou, C.; Pang, W. Particle swarm optimization for traveling salesman problem. In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, Xi'an, China, 5 November 2003; Volume 3, pp. 1583–1585.
29. Xu, X.; Li, J.; Zhou, M. Bi-objective colored traveling salesman problems. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 6326–6336. [[CrossRef](#)]
30. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]

31. Liu, R.; Du, B.; Zhang, L. Multiobjective optimized endmember extraction for hyperspectral image. *Remote Sens.* **2017**, *9*, 558. [[CrossRef](#)]
32. Lu, K.; Wu, Z. Multi-Objective False Data Injection Attacks of Cyber–Physical Power Systems. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *69*, 3924–3928. [[CrossRef](#)]
33. Wei, L.; Cai, Z.; Zhou, K. Multi-objective Gray Wolf Optimization Algorithm for Multi-agent Pathfinding Problem. In Proceedings of the 2022 IEEE 5th International Conference on Electronics Technology (ICET), Chengdu, China, 13–16 May 2022.

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.