

Article

# Scheduling of Multi-AGV Systems in Automated Electricity Meter Verification Workshops Based on an Improved Snake Optimization Algorithm

Kun Shi <sup>1</sup> , Miaohan Zhang <sup>1</sup>, Zhaolei He <sup>2</sup>, Shi Yin <sup>1,\*</sup>, Zhen Ai <sup>3</sup> and Nan Pan <sup>1</sup> 

<sup>1</sup> Faculty of Civil Aviation and Aeronautical, Kunming University of Science and Technology, Kunming 650500, China; 20222245026@stu.kust.edu.cn (K.S.); 20212245025@stu.kust.edu.cn (M.Z.); nanpan@kust.edu.cn (N.P.)

<sup>2</sup> Metering Center, Yunnan Power Grid Co., Ltd., Kunming 650200, China; hezhaolei@yn.csg.cn

<sup>3</sup> NARI Nanjing Control System Co., Ltd., Nanjing 211100, China; aizhen@sgepri.sgcc.com.cn

\* Correspondence: shi.yin@kust.edu.cn

**Abstract:** Automated guided vehicles (AGVs) are one of the core technologies for building unmanned autonomous integrated automated electric meter verification workshops in metrology centers. However, complex obstacles on the verification lines, frequent AGV charging, and multi-AGV collaboration make the scheduling problem more complicated. Aiming at the characteristics and constraints of AGV transportation scheduling for metrology verification, a multi-AGV scheduling model was established to minimize the maximum completion time and charging cost, integrating collision-avoidance constraints. An improved snake optimization algorithm was proposed that first assigns and sorts tasks based on AGV-order-address three-level mapping encoding and decoding, then searches optimal paths using an improved A\* algorithm solves multi-AGV path conflicts, and finally finds the minimum-charging-cost schedule through large neighborhood search. We conducted simulations using real data, and the calculated results reduced the objective function value by 16.4% compared to the traditional first-in-first-out (FIFO) method. It also reduced the number of charges by 60.3%. In addition, the proposed algorithm is compared with a variety of cutting-edge algorithms and the results show that the objective function value is reduced by 8.7–11.2%, which verifies the superiority of the proposed algorithm and the feasibility of the model.

**Keywords:** logistics applications; multi-AGV scheduling; charging constraints; improved snake optimization algorithm; improved A\* algorithm path planning



**Citation:** Shi, K.; Zhang, M.; He, Z.; Yin, S.; Ai, Z.; Pan, N. Scheduling of Multi-AGV Systems in Automated Electricity Meter Verification Workshops Based on an Improved Snake Optimization Algorithm. *Symmetry* **2023**, *15*, 2034. <https://doi.org/10.3390/sym15112034>

Academic Editor: László T. Kóczy

Received: 18 October 2023

Revised: 4 November 2023

Accepted: 6 November 2023

Published: 8 November 2023



**Copyright:** © 2023 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

In automated electric meter verification workshops, automated guided vehicles (AGVs) autonomously navigate and transport various types of smart electric meters to designated verification stations for loading and unloading operations using path planning algorithms, such as the A\* algorithm and Dijkstra's algorithm, according to the verification process flow and meter types [1–3]. AGVs can precisely deliver meters to specified stations in time following the processing and verification sequence requirements, thanks to their advantages of continuous operation, precise docking, and flexible routing, which significantly improve transportation efficiency and reduce labor costs. Based on the multi-AGV scheduling problem in automated electric meter verification workshops, this study adds constraints like battery capacity, dynamic charging demands, and charging station resources in order to meet the needs of collaborative scheduling and power optimization of multiple AGVs. Therefore, this is an electric vehicle routing problem with time windows (EVRPTW) [4–6]. With the growing demand for automated meter verification from electric utilities, the number of AGVs and the scale of transportation tasks also increase substantially in automated electric meter verification workshops. Thanks to advantages such as short computation

time and minimal problem requirements, heuristic algorithms, including genetic algorithm, ant colony algorithm, and particle swarm optimization have been widely applied to AGV path planning problems and achieved certain progress [7–10].

However, traditional heuristic algorithms still have limited performance in dealing with large-scale AGV scheduling problems under multi-dimensional constraints. Their random search can easily get stuck in local optima, and it is difficult to effectively implement efficient scheduling and dynamic autonomous path planning for AGV tasks considering the multi-dimensional constraints in automated electric meter verification workshops [11,12]. Against this background, how to effectively reduce the scheduling and path costs of large-scale AGVs based on complex dynamic environments to maximize the overall productivity of the verification system has become an urgent issue that electric power enterprises need to resolve.

On one hand, some scholars usually treat large-scale dynamic AGV scheduling problems as several sub-problems to study respectively. The literature [13] decomposes the large-scale AGV scheduling and path planning problem into upper-level task allocation sub-problems and lower-level path planning sub-problems based on hierarchical planning ideas to reduce problem complexity. The literature [14] mathematically characterized the AGV scheduling problem in flexible job shops and designed a hybrid algorithm in steps to achieve flexible coordination between AGVs for transportation tasks and different processing machines. With technological advances, the carrying capacity of AGVs has been significantly improved, enabling them to conduct bulk transportation and loading/unloading operations. This poses higher requirements on AGV scheduling and path planning algorithms, which need to fully consider multi-constraint conditions like AGV load constraints, loading/unloading time constraints, etc., and conduct more sophisticated transportation planning. Based on real-time dispatching strategies of AGVs, the literature [15,16] proposed a multi-task chain scheduling algorithm considering the remaining cargo capacity characteristics of AGVs and validated through a real smart manufacturing system that it can significantly reduce scheduling costs. For multi-load AGV scheduling problems with capacity constraints, the literature [17] proposed an improved ant colony optimization-simulated annealing algorithm based on multi-attribute scheduling rules, but without considering AGV quantity limits. Further taking into account that the number of AGVs is limited, the literature [18] proposed an improved genetic algorithm to solve flexible job shop scheduling problems with multiple AGVs. The literature [19] balanced the distribution of traffic loads for all AGVs executing tasks in the network to avoid local congestion. However, the above literature did not consider the transportation characteristics of carrying goods and production processes, thus lacking algorithm practicality. The literature [20] designed an integrated coding method for production plans and process routes, solving AGVs' process selection, operation sequence, and transportation task assignment problems, but did not propose an efficient algorithm to solve this problem.

On the other hand, it is difficult to coordinate and optimize paths between different AGVs in real-time, especially in complex transportation environments. The larger the scale of AGVs, the higher the possibility of collisions, system deadlocks, and road congestion and blockage. These system uncertainties caused by multi-AGV scheduling will severely reduce the overall transportation efficiency of automated electric meter verification workshops. The literature [21–24] explicitly considered the bidirectional path conflicts of AGVs to formulate mixed integer programming (MIP) models for avoiding vehicle collisions, with the goal of minimizing the completion time of all tasks. The literature [25] further studied the path planning and task allocation problems when two types of AGVs (horizontal or vertical handling) are mixed and utilized A\* algorithms combined with cyclic planning to achieve the shortest collision-free paths between any two points and fixed obstacles. The literature [26] proposed a new deadlock avoidance control strategy for the deadlock problem when AGVs handle parts during production and transportation. However, the above literature did not study real-time coordination mechanisms between AGVs. To avoid collisions, deadlocks, and congestion in dynamic multi-AGV environments, it is

necessary to propose core algorithms for AGV motion decision-making under complex dynamic environments to ensure efficient collaborative completion of transportation tasks by multiple AGVs.

In addition, AGVs themselves have battery capacity limitations. After continuous heavy-duty operation, battery endurance will severely restrict AGV transportation efficiency. The literature [27–29] assigned transportation and charging requests to AGVs while determining their start times and AGV charging durations and established mixed integer linear programming models to minimize transportation delays. On this basis, the literature [30] addressed multi-load and multi-function AGV collaborative scheduling problems through battery management and solved charging request schedules based on the proposed hybrid adaptive large neighborhood search. However, the above literature did not study joint optimization methods for charging strategies and transportation scheduling. How to fully consider battery power constraints during transportation task planning and scheduling to minimize the impact of charging on transportation efficiency is also a key difficulty in current research.

Regarding the existing literature on solving AGV scheduling problems within automated electric meter verification workshops, Table 1 summarizes the differences between this paper and related studies in addressing this type of problem, with an X mark indicating that the literature focuses on a certain aspect or what method is used. Based on the operation process of the automated meter verification workshop, this paper fills this research gap by considering the capacity of multiple AGVs, charging requests, collaborative obstacle avoidance between AGVs, and path optimization. To minimize the number and time of charging, this study optimized the sequence of charging tasks based on the electric vehicle routing problem with the time windows model and conducted collaborative path planning for multiple vehicles to avoid collisions. From the perspective of global coordination of charging tasks, this study proposed a new collaborative method between AGV charging and operations, providing a new perspective to reduce long-cycle operating costs of the system and ensure operational safety.

In summary, our contributions are:

- (1) Based on the complex environment of the automated meter verification workshop, we consider the multi-dimensional factors such as AGV path planning, collaborative obstacle avoidance, and charging constraints, and construct an AGV scheduling model for the automated meter verification workshop, considering the charging constraints with the goal of minimizing the distribution cost and charging cost.
- (2) For large-scale order scheduling and multi-AGV scheduling problems, we designed an improved snake optimization algorithm (ISO). By adopting a random reverse learning strategy to generate a larger initial population solution space followed by individualized memory strategy and Gaussian mutation operations to enhance global and local search capabilities, significant improvements in solving efficiency and solution quality were achieved. Experimental results show that the proposed algorithm can greatly reduce the total cost required for AGVs to complete orders compared to traditional methods.

For the frequent charging problem of AGVs, we designed a charging optimization method based on a large neighborhood search strategy. By changing the charging plan through different destruction and repair operators, this method reduced the total number of chargings and charging costs. For collision conflicts during AGV navigation, we improved the A\* path planning algorithm. By introducing direction change penalty and anti-collision mechanisms, the continuity of the global path was optimized.

- (3) In the simulation experiments, we use algorithm benchmarking, case analysis, and parameter sensitivity analysis to comprehensively verify the effectiveness of the designed method in dealing with high-dimensional optimization problems, reducing the path length of AGVs, and reducing charging costs. In addition, the algorithm process is analyzed through a visual method and the effectiveness of the proposed strategy is visually demonstrated.

Table 1. Current and previous studies.

	M. Mousavi (2017) [7]	J. Li (2022) [31]	A. Ammar (2016) [2]	M. Gen (2014) [10]	X. Shao (2023) [11]	J. C. Tu (2017) [12]	E. Hu (2023) [13]	B. Xin (2023) [14]	H. Niu (2023) [15]	X. A. Martin (2023) [16]	Z. Wang (2023) [17]	L. Meng (2023) [18]	Y. Gao (2023) [19]	Q. Liu (2023) [20]	Q. Zeng (2022) [21]	T. Adamo (2023) [22]	B.-H. Zhou (2023) [23]	W.-Q. Zou (2020) [24]	Z. Jiang (2023) [25]	N. Wu (2005) [26]	N. Singh (2022) [27]	M. Boccia (2020) [28]	Q.-V. Dang (2021) [30]	Current work	
<b>VRP</b>																									
multi-vehicle	x		x	x	x	x	x	x	x		x		x	x		x			x	x	x	x	x	x	x
single vehicle		x										x			x		x	x							
Pickup and delivery	x			x	x	x		x					x	x		x									x
Capacitated	x						x	x	x		x	x	x			x		x			x				x
<b>Vehicle fleet</b>																									
Homogeneous	x	x		x		x		x	x	x	x	x	x	x		x	x	x							x
Heterogeneous			x		x		x																		
<b>Time windows</b>																									
None	x	x	x	x	x	x	x	x	x	x		x	x	x			x	x		x		x	x	x	
Soft											x										x				x
Hard															x	x			x						
<b>Battery management</b>																									
None		x		x	x	x		x	x	x	x	x	x	x		x				x					
Battery swapping																									
Full charge							x																		
Partial Charge	x				x																				
<b>Search method</b>																									
LNS														x											
Exact method	x				x	x											x								
RL										x			x		x										
Heuristic algorithm	x	x	x	x			x	x	x		x	x		x			x								
<b>Route planning</b>																									
None	x			x	x		x	x				x													
A-star algorithm		x	x																						
Crashworthiness						x				x	x	x					x	x							
Collaboration									x		x		x	x		x									

## 2. Materials and Methods

### 2.1. Background

The traditional material scheduling of the automated meter verification workshops adopts the principle of first-in-first-out (FIFO) and the orders are assigned to each AGV in turn according to the order acceptance time; the AGV completes the process of pick-up and delivery in a point-to-point manner after receiving the order request. This scheduling method does not fully consider the route and charging time of AGVs, is limited to the optimization of a single order route, and ignores the overall optimization of multiple tasks of AGVs in parallel, which usually causes the waste of AGV routes and unnecessary charging time, making the overall work efficiency low and the operating costs too high.

The AGV fleet scheduling problem for the automated meter verification line shown in Figure 1 is described as follows: The automated electric meter verification workshops have  $k$  ( $k \in K$ ) AGVs and  $q$  ( $q \in Q$ ) verification electric meter materials orders that need to be transported. The verification line process, based on order data within a given period and the current inventory status, schedules the transport robots to move the required metering materials from the warehousing location for weekly packaging to the designated verification line entrance (i.e., outbound tasks), and when the remaining power of the AGVs is lower than a certain value or unable to complete the next task, they need to recharge at the charging area to continue the transport work (i.e., charging tasks). When the remaining power of the AGV is lower than a certain value or when it cannot complete the next task after picking, it needs to enter the charging area to restore the power so that it can continue the handling work. The robot AGV carries a crate containing three kinds of metering materials: single-phase energy meters, three-phase energy meters, and low-voltage transformers. The measuring material storage area totals 3 areas, a single area of 5 rows, 5 boxes, and 4 groups, that is, 100 stacks. Each stacks holds a maximum of 18 crates, so the maximum number of crates in a single area is 1800, in addition, each crate

can carry up to 12 meters. AGVs can execute path commands to move from node  $m$  to  $n$ , and each step can only be moved to 8 map nodes connected around the center of the AGV. AGVs go to the address of the goods location of the materials in the order, and there are stacking stackers on the shelves that give the goods to the AGVs in advance for moving them to one of the corresponding inspection line entrances. The AGV goes to the address of the goods position in the order. After entering the entrance of the inspection line, if there are obvious defects in the appearance of the materials, they will be sorted in the abnormal processing area. In addition, unqualified products will also enter the abnormal processing area after passing inspection, while qualified products will be returned to the warehouse or shipped to perform the process.

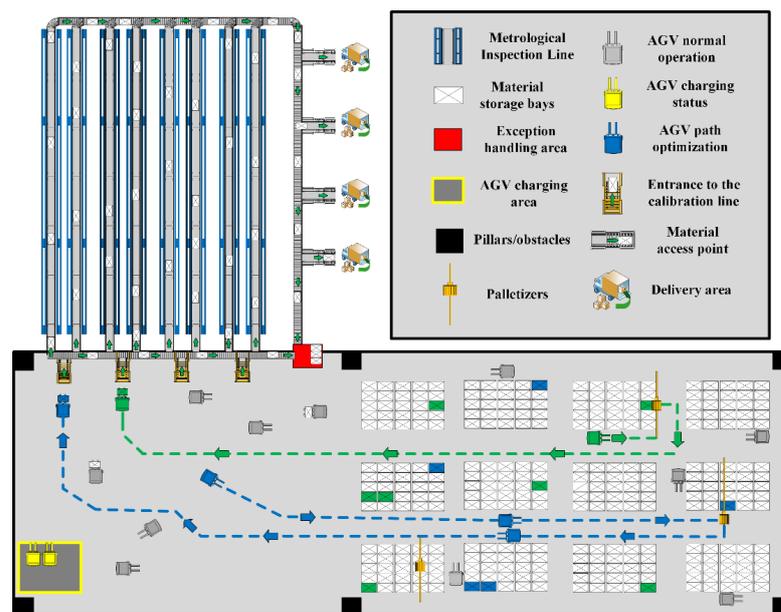


Figure 1. Automated electric meter verification workshops.

The core of the automated verification line is to coordinate and optimize the execution of the above tasks to maximize the efficiency of warehousing, the arrangement of AGV tasks and routes needed to meet the global optimal, achieve real-time response, and avoid congestion/deadlock, etc., and at the same time balancing the workload between each region and each AGV, in addition to the need to reasonably arrange charging tasks to avoid too many AGVs charging in a period that affects the efficiency of the work. In this paper, based on the scenarios of the AGV's stocking and charging tasks in the automated verification line, a model is established with the optimization objective of minimizing the maximum handling completion time while at the same time considering multi-dimensional constraints of the AGV's power loss, collision prevention, avoidance of congestion and deadlocks during the working process. Optimal handling order sequence, handling path, and charging plan of the AGVs are solved to achieve the improvement of the overall work efficiency.

## 2.2. Model Assumptions

1. We assume that the automated verification line materials fully meet the order demand, and only consider the AGVs out of the warehouse task and charging task scenarios;
2. We assume the AGV reaches the pickup location and materials are loaded onto it, without accounting for the time needed for loading.
3. We assume that the AGVs travel at a constant speed during the whole process, the power loss of the AGV has a linear relationship with the transportation distance, i.e., ignoring the start-stop time of the AGVs;

4. When AGVs work, the discharge current is continuous and stable, ignoring the impact of power loss caused by the load of metering materials;
5. Neglecting the AGV charging process by increasing the charging time window penalty required to complete the charging task.
6. In previous studies, the paths of individual AGVs were considered to be asymmetric due to the influence of various factors such as orders, complex scenarios, and algorithm differences [31]. To study the path planning and motion control of AGVs in complex factory environments, this paper abstracts and models an automated electric meter verification workshop as a grid map. In the map (Figure 2), white grids represent traversable areas, with each grid representing a certain distance unit; blue grids denote shelves for storing materials; black grids are obstacle pillars; red grids indicate emergency handling areas; yellow grids represent charging stations; pink grids denote verification workshop platforms; and green grids represent material transfer points. Among them, pillars, emergency handling areas, platform areas, and the interiors of shelves are defined as obstacle areas. When an AGV sends a charging request, the system will randomly assign an available charging station, which is then no longer considered an obstacle. The raster map consists of  $56 \times 23$  grids, in which a Cartesian coordinate system is established from the lower left corner, and each grid is represented by coordinates (X,Y), where X is between 1 and 56 and Y is between 1 and 23.

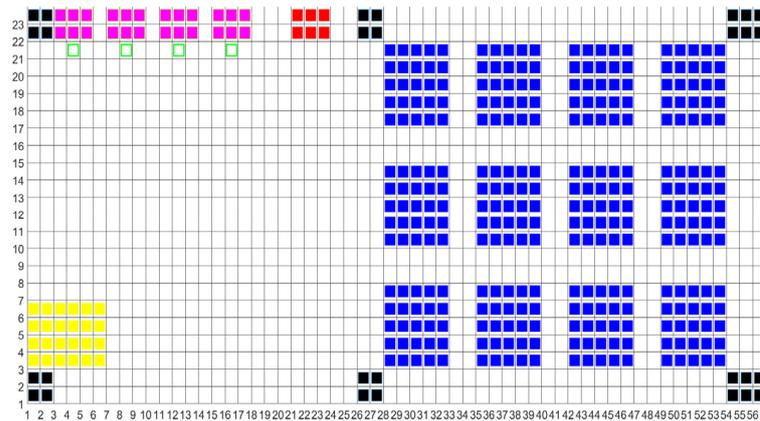


Figure 2. Raster map of automated electric meter verification workshops.

### 2.3. Symbol Description

Table 2 provides a list of the variables and parameters used in the model. This overview of the key model components and their attributes will aid comprehension of the model’s structure and characteristics. Table 3 illustrates the state variables in the model, which typically contain the numbers 0 and 1.

Table 2. Description of the main mathematical symbols.

Parameter	Symbol Description
$Q$	All order collections
$V$	The set of AGV-feasible nodes
$W$	Collection of all material storage spaces
$I$	Collection of material types
$P$	All verification line entrances
$K$	Collection of all AGVs
$T$	Collection of all time points on the timeline
$q$	Orders, $q \in Q$
$w$	material storage space, $w \in W$
$i$	Measuring supplies, $i \in I$
$p$	Entrance to the verification line $p \in P$

Table 2. Cont.

Parameter	Symbol Description
$k$	Handling robot AGV, $k \in K$
$t$	Time
$N_{wi}$	Quantity of $i$ dosing material in the storage space $w$
$q_{wp}$	Transportation distance from the storage space $w$ to the verification line entrance $p$
$d_{pi}$	Demand for metering materials $i$ at the entrance to the verification line $p$
$y_{wpi}$	Quantity of metered materials sent from storage to verification line entrance
$N_{AGV}^t$	Number of free AGV carts at any given time
$\bar{N}$	Average number of orders assigned to each picking station
$N$	Number of picking stations
$e_{mn}$	Distance from $m$ node to node $n$
$\gamma$	AGV traveling speed
$E_{k-max}$	AGV battery full charge
$S_{k-total}$	Total AGV running time
$E_{BL}$	Minimum amount of power needed to charge the AGV
$Q_k$	Charging time of the $k$ th unit
$V_C$	Charging rate of the AGV
$E_{k-now}$	Actual power consumption of the $k$ th AGV
$E_{k-remain}$	Power consumed by the $k$ th AGV

Table 3. 0–1 variables and their descriptions.

Variables	Symbol Description
$X_{wp}$	$X_{wp} = \begin{cases} 1, & \text{The storage level } w \text{ is shipped to the inlet } p \text{ of the calibration line.} \\ 0, & \text{otherwise} \end{cases}$
$x_{wp}^{k,t}$	$x_{wp}^{k,t} = \begin{cases} 1, & \text{Material transported from storage position } w \text{ to the entrance } p \\ & \text{of the calibration line by trolley } k \\ 0, & \text{otherwise} \end{cases}$
$X_{mnk}^t$	$X_{mnk}^t = \begin{cases} 1, & \text{AGV } k \text{ cart from node } m \text{ to node } n \text{ in time } t \\ 0, & \text{otherwise} \end{cases}$

#### 2.4. Multi-AGV Scheduling Model of Automated Meter Verification Workshop Considering Charging Constraints

$$\min f = f_1 + f_2 \quad (1)$$

$$f_1 = \sum_{t \in T} \sum_{k \in K} \sum_{p \in P} \sum_{w \in W} q_{wp} \cdot x_{wp}^{k,t} \quad (2)$$

$$f_2 = \sum_{k \in K} Q_k \cdot \lambda C_k \quad (3)$$

Equation (1) represents the minimizing AGV transport distance and charging cost as an objective function; Equation (2) represents the maximum transport distance for all AGVs; Equation (3) represents the charging cost for all AGVs. The  $\lambda$  is the number of charge weights and  $C_k$  is the charging times. The constraints are:

$$\sum_{w \in W} y_{wpi} = d_{pi}, \forall i \in I, p \in P \quad (4)$$

$$0 \leq \sum_{p \in P} y_{wpi} \leq N_{wi}, \forall i \in I, w \in W \quad (5)$$

$$\sum_{i \in I} y_{wpi} \leq x_{xp} \cdot \sum_{i \in I} d_{pi}, \forall w \in W, p \in P \quad (6)$$

$$x_{wp} \in \{0, 1\}, \forall p \in P, w \in W \quad (7)$$

$$y_{wpi} \geq 0 \quad (8)$$

$$\sum_{k \in K} \sum_{w \in W} X_{wp}^{k,t} \leq b, \forall t \in T, p \in P \quad (9)$$

$$\sum_{k \in K} X_{wp}^{k,t} \leq b, \forall t \in T, w \in W, p \in P \quad (10)$$

$$\sum_{i \in I} \sum_{w \in W} N_{wi} \cdot X_{wp} \geq \sum_{i \in I} d_{pi}, \forall p \in P \quad (11)$$

$$\bar{N}(1 - \delta) \leq \sum_{i \in I} \sum_{w \in W} \frac{N_{wi} \cdot X_{wp}}{\sum_{p \in P} \sum_{i \in I} \frac{d_{pi}}{\bar{N}}} \leq \bar{N}(1 + \delta), \forall p \in P \quad (12)$$

$$\sum_{k \in K} \sum_{n \in V} X_{mnk}^t = 1, n \in V, t \in T \quad (13)$$

$$\sum_{k \in K} \sum_{n \in V} X_{mnk}^t = 1, m \in V, t \in T \quad (14)$$

$$e_{mn} \neq e_{nm} \quad (15)$$

$$t_{mn} = \frac{e_{mn}}{\gamma}, m, n \in V \quad (16)$$

$$T_{mnk}^{in} + t_{mn} + t_{mnk}^{wait} \leq T_{n+1, m+1}^{in}, \forall m, n \in V, k \in K \quad (17)$$

$$\left| T_{mnk}^{in} - T_{mnk'}^{in} \right| < \frac{H_{safe}}{\gamma}, \forall m, n \in V; k, k' \in K \quad (18)$$

$$Q_k = \frac{E_{k-max} - E_{k-now}}{V_C} \quad (19)$$

$$\begin{cases} E_{k-max}, E_{k-now} \leq E_{BL} \\ E_{k-now}, E_{k-now} > E_B \end{cases} \quad (20)$$

Equation (4) indicates that the quantity of goods shipped to the entrance of the verification line  $p$  is equal to the order demand; Equation (5) indicates that the quantity of material delivered to the entrance of the verification line  $p$  from a storage space  $w$  is less than or equal to the quantity of material in each storage space; Equation (6) indicates whether the commodity at the entrance of the verification line  $p$  is supplied by the storage position  $w$ ; Equation (7) is the 0–1 variable constraint, which is 1 when the pallet is shipped to the checkpoint entrance and 0 otherwise; Equation (8) is the transportation quantity constraint. Equation (9) means that the number of AGVs arriving at the checkpoint entrance at any moment should be less than the maximum storage capacity of the checkpoint entrance; Equation (10) means that each storage space is transported by only one AGV at any moment. Equation (11) indicates that the quantity of material in the storage space  $w$  meets the demand for material  $i$  at the verification line  $p$ ; Equation (12) indicates that the deviation of the order from the average distribution is within  $\delta$ .  $N_{wi}$  indicates the number of commodities owned by each pallet; Equations (13) and (14) indicate that there is at most one AGV traveling through any node at time  $t$  to avoid the occurrence of AGV node conflicts; Equation (15) indicates that the path network that can be exercised by AGVs is a unidirectional guided path; Equation (16) represents the time that the AGV passes through

the path node  $m \rightarrow n$  segment; Equation (17) represents the continuity of AGV trolley transportation time; Equation (18) represents the need to maintain a safe time interval when entering the same path node through any two trolleys. Equation (19) represents the charging duration of the AGV; Equation (20) represents the AGV charging operation, i.e., the current charge is charged if it is less than or equal to the minimum charge and vice versa.

### 3. Algorithm Description

#### 3.1. Standard SO Algorithm

AGV cluster scheduling for inspection line needs to optimize the objectives such as job scheduling order of AGVs, transportation route planning, etc. In addition, there are complex situations such as path or node conflicts between multiple AGVs as well as consideration of coordinated charging orders. The multi-AGV scheduling optimization problem has many variables and constraints and the computational complexity of solving the optimal solution is high, which requires the design of efficient optimization algorithms. Snake Optimizer (SO) is an optimization algorithm proposed by Professors Hashim, F. A. and Hussien, A. G in 2022, whose algorithm simulates the foraging and reproduction behavior of snakes [32]. Since it has the advantages of few parameters and good global search performance, and it has been proven to have strong advantages in engineering optimization and multi-objective optimization problems, solving this kind of optimization problem by SO has a broader application prospect.

The optimization process for the SO algorithm is divided into the exploration phase and the development phase. The exploration phase simulates the behavioral patterns of snakes in the absence of food, and the exploitation phase simulates the behavioral patterns of snakes in the presence of food, which is controlled by the total amount of physical objects  $Q$  and temperature  $Temp$ .

If  $Q < Threshold_Q(0.25)$ , snakes find food by choosing any random location and updating their position. The position update formula is as follows:

$$X_i(t+1) = X_{rand}(t) \pm X_i^{EP1} \quad (21)$$

$$X_i^{EP1} = c_2 \times A \times ((X_{max} - X_{min}) \times rand + X_{min}) \quad (22)$$

where  $X_i$  denotes the position of the  $i$ -th individual (male or female),  $X_i^{EP1}$  denotes the search distance of the  $i$ -th individual at that stage,  $X_{rand}$  denotes that an individual is randomly selected,  $A$  denotes the predatory ability of the  $i$ -th individual,  $rand \in (0, 1)$ ,  $c_2 = 0.05$ ,  $t$  represents the current number of iterations, and  $X_{min}$  and  $X_{max}$  represent the lower and upper boundaries of the population, respectively.

If  $Q > Threshold_Q$  and  $Temp > Threshold_{Temp}(0.6)$ , the snake will only move toward the food. The equation for the movement of an individual snake is as follows:

$$X_i(t+1) = X_{food} \pm X_i^{EP2} \quad (23)$$

$$X_i^{EP2} = c_3 \times Temp \times rand \times (X_{food} - X_i(t)) \quad (24)$$

where  $X_i$  is the position of the  $i$ -th individual,  $X_i^{EP2}$  denotes the search distance of the  $i$ -th individual at that stage,  $X_{food}$  is the position of the optimal individual, and  $c_3$  is constant and equal to 2.

It is worth noting that the snake optimization algorithm divides the population into two parts, male and female, with males fighting to gain the opportunity to mate with females, and females producing new individuals through mating. If  $Temp < Threshold_{Temp}(0.6)$ , the snake is in fighting mode or mating mode and the equation for the movement of an individual snake in fighting mode is as follows:

$$X_i^{MM}(t+1) = X_i^{MM}(t) + X_i^{EP3} \quad (25)$$

$$X_i^{EP3} = c_3 \times MM \times rand \times (Q \times X_{MM}^{food} - X_i^{MM}(t)) \quad (26)$$

where  $X_i^{MM}$  is the position of the  $i$ -th male individual,  $X_i^{EP3}$  denotes the search distance of the  $i$ -th individual at that stage,  $X_{MM}^{food}$  is the best position in the male group, and  $MM$  is the male fighting strength.

If  $Temp < Threshold_{temp}(0.6)$ , the equation for the movement of an individual snake in mating mode at the time is as follows:

$$X_i^{FM}(t+1) = X_i^{FM}(t) + X_i^{EP4} \quad (27)$$

$$X_i^{EP4} = c_3 \times FM \times rand \times (Q \times X_{FM}^{food}(t) - X_i^{FM}(t)) \quad (28)$$

where  $X_i^{FM}$  is the position of the  $i$ -th female individual,  $X_i^{EP4}$  denotes the search distance of the  $i$ -th individual at that stage,  $X_{FM}^{food}$  is the best position in the female group, and  $FM$  is the mating ability of the female individual.

### 3.2. Improved SO Algorithm

#### (1) Random opposition-based learning initialization

In order to obtain a better quality initial solution, a stochastic inverse learning strategy is designed in this section. During the process of population optimization, an inverse solution is generated based on the current solution, comparing the objective function values of the current solution and the inverse solution and selecting the best one to enter the next iteration.

$$\tilde{X}_{i,j} = LB + UB - X_{i,j} \quad (29)$$

where  $LB$  represents the minimum value of the problem variable,  $UB$  represents the maximum value of the problem variable, and the problem variable of the article is set to the order set  $Q$ .  $X_{i,j}$  represents the randomly generated population. Here,  $i$  represents the number of populations, and  $j$  represents the number of problem variables.  $\tilde{X}_{i,j}$  represents the generated inverse learning initial population. Since the reverse solution generated by the reverse learning strategy has a certain value of distance from the current solution, it lacks randomness and cannot effectively enhance the population diversity in the search space. Therefore, the literature [33] proposed an improved Random Opposition-based Learning (ROBL) strategy to further enhance the population diversity and avoid the ability to fall into a local optimum, which is calculated as follows:

$$\tilde{X}_{rand} = LB + UB - r \times X_{i,j} \quad (30)$$

where  $\tilde{X}_{rand}$  denotes a random reverse solution and  $r$  denotes a random number between 0 and 1.

#### (2) Individualized memory strategy

In the basic SO algorithm, the update of the optimal position relies on dividing the population into female and male individuals for updating. The movement and convergence of individuals in the search space to obtain the optimal solution are achieved through the fighting of the male population and the mating of the female population. However, the basic SO algorithm does not consider the value of the eliminated individuals themselves in the fighting mode. Therefore, based on the fighting mode, the individual memory function in the particle swarm optimization algorithm is considered and its specific expression is:

$$X_i^{MM}(t+1) = b_1 \cdot \sum_{j=1} X_i^{MM}(t+1) + b_2 \cdot rand \cdot (P_{best}^{MM} - X_i^{MM}(t)) \quad (31)$$

where,  $rand$  represents a random variable between  $[0,1]$ ,  $b_1$  and  $b_2$  represent the group combat coefficient and the eliminated individual memory coefficient, respectively, which are constants between  $[0,1]$ .  $P_{best}^{MM}$  represents the best position experienced by the  $i$ -th male snake individual. By adjusting the value of  $b_1$  and  $b_2$ , the influence of group communication and individual memory on the search can be balanced.

### (3) Gaussian variational strategy

The mutation operator can avoid the algorithm falling into local optima while maintaining the diversity of the population. In order to reduce the probability of premature convergence and getting stuck in local optima in the basic SO algorithm, this paper performs a Gaussian mutation operation on the current optimal solution  $X_{worst}^{FM}$  with a certain probability  $P$  and adopts a “greedy” selection idea to implement the selection rule of survival of the fittest.  $X_{worst}^{FM}$  represents the individual with the least fitness in the female population. The specific expression of the Gaussian mutation operator is as follows:

$$X_G^{FM}(t+1) = X_{worst}^{FM}(t)(1 + Gaussian(\sigma)) \quad (32)$$

$X_G^{FM}$  represents the position of the mutated individual,  $Gaussian(\sigma)$  is a random variable following Gaussian distribution. The update of the global optimal position is as follows:

$$X_i^{FM}(t+1) = \begin{cases} X_i^{FM}(t+1), & otherwise \\ X_G^{FM}(t+1), & f(X_G^{FM}(t+1)) < f(X_i^{FM}(t+1)) \text{ and } rand < P \end{cases} \quad (33)$$

$rand$  represents a random variable between  $[0,1]$ ,  $P$  is the probability of survival of the fittest selection, and  $f(\cdot)$  is the fitness value of the individual. By mutating the current global optimal solution  $X_{FM}(t)$ , getting stuck in local optima can be avoided (in case the current global optimum is a local optimum). Adopting this selection strategy enables the population to evolve toward the optimal solution while effectively improving the search efficiency of the algorithm.

The red border in Figure 3 delineates the proposed ISO algorithm flow, while the yellow border indicates the improved A\* algorithm. Initially, all orders are assigned to AGVs using ISO. The objective function is then computed via the improvement A\* algorithm. The iterative process terminates once the maximum number of iterations is reached, upon which the optimal AGV scheduling plan is generated.

### 3.3. Improved A\* Algorithm

The A\* algorithm is used to solve for the shortest path in the plane and is a typical heuristic algorithm [34]. This paper improves the A\* algorithm through the following three steps, allowing it to be better applied to solving the AGV fleet scheduling model for automated electric meter verification workshops.

Step 1: In the A\* algorithm, the heuristic function is the most critical component that guides the search direction and determines the efficiency of the algorithm. To reduce the number of turns of AGVs during transportation, this study designed a heuristic function that penalizes turns. Specifically, when estimating the total cost of a path, if the path involves turns, an additional penalty value is added to the cost. This will make the A\* algorithm prefer paths with fewer turns during the search process. Experiments show that compared to the default heuristic, this turn-penalizing heuristic function can effectively reduce the number of turns of AGVs, making the transportation trajectories smoother and more continuous, achieving the goal of simplifying transportation paths and improving transportation efficiency. The co-equation of the A\* algorithm is expressed as:

$$f(a) = g(a) + h(a) + C(a) \quad (34)$$

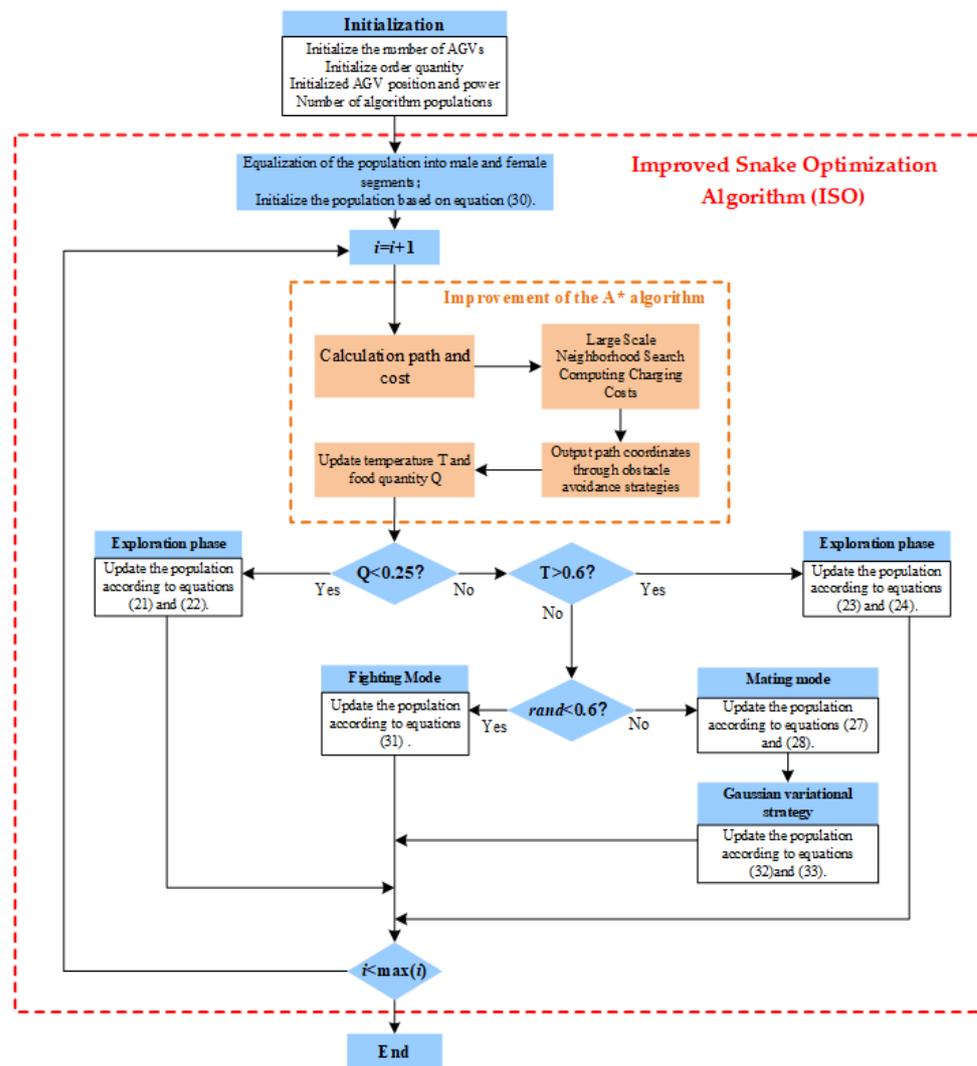


Figure 3. Implementation of the ISO algorithm.

In Equation (34),  $f(a)$  is the total cost of the current node,  $g(a)$  is the distance traveled from the starting point  $s$  to the current node  $a$ , and  $h(a)$  is the Manhattan distance from the current node to the endpoint. The formula for the extra cost  $C(a)$  is as follows:

$$C(a) = \begin{cases} 0 & ,if X_P = X_N or Y_P = Y_N \\ f(a) & ,if X_P = X_N and Y_P = Y_N \end{cases} \quad (35)$$

The Manhattan distance formula is as follows:

$$d = |x_1 - x_2| + |y_1 - y_2| \quad (36)$$

Step 2: Arrange the points to be explored according to the cost from smallest to largest, take the point with the smallest cost as the target point of the next round, i.e., the starting point of the next round, and delete the point from the unexplored array and add it to the explored array. The third step is to record the path from the current node  $a$  to the starting point  $s$ , determine whether the current node is the end point. If not, repeat step 1 until the endpoint is reached and terminate the loop.

The main search directions of the A\* algorithm are 4-way search and 8-way search. The search directions for the 4-way search are up, down, left, and right, and the motion step of the AGV is 1 at each time, while the search directions for the 8-way search are up,

down, left, right, left up, left down, right up, and right down, and the motion step of the robot is  $\sqrt{2}$ .

Step 3: To prevent collisions between AGVs during transportation tasks, this paper incorporated a collision avoidance strategy into the A\* path planning algorithm. The strategy categorizes AGV collision scenarios into two types: head-on collisions and non-head-on collisions. Considering that the vehicles move at the same speed, rear-end collisions are not considered. In head-on collision scenarios, vehicles with heavier loads and higher overall costs are assigned priority of way, while the other vehicles make a turn to give way. In non-head-on scenarios, the priority vehicles continue straight, while the others wait before reaching the conflict point to avoid collision. Figure 4 illustrates (a) head-on collisions and (b) non-head-on collisions. In (a),  $\times$  represents the conflict between two AGVs in the upper portion, while the algorithm below calculates the assigned right-of-way priority. The AGV then detours to avoid the collision, indicated by  $\surd$ , showing the conflict has been resolved and the AGV can proceed normally. Similarly, the upper portion of (b) shows a non-head-on collision occurring. After the algorithm computes the right-of-way priority in the lower half, one AGV successfully avoids the collision by employing a waiting strategy.

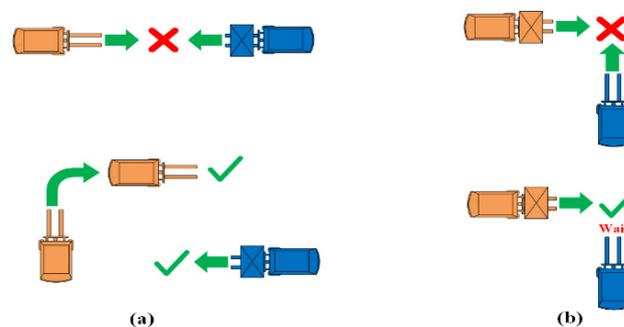


Figure 4. (a) Head-on collisions; (b) non-head-on collisions.

### 3.4. AGV Encoding and Decoding

In order to design the mapping relationship between the scheduling model and the algorithm, and then realize the efficient application of the algorithm, a coding mapping relationship based on AGV-order shelf storage space is designed by comprehensively taking into account the actual meter distribution situation. In the encoding process, the number of orders is the dimension of the problem, and the AGV number is the variable used to generate the initial order allocation population; the optimal combination of AGVs and orders is realized by updating the mapping relationship between different orders and AGV numbers. Specific examples of decoding and mapping relationships are shown below: Assuming that the total number of AGVs in the metering and verification line  $p$  ( $p \in P$ ) undertakes the number of distribution orders for  $Q$ , the task number is three, corresponding to the type of order meters for three kinds of power metering materials  $\{A, B, C\}$ .  $A$  represents the single-phase meter,  $B$  represents the three-phase meter, and  $C$  represents the mutual inductor. The number of meters required by the order is  $\{8, 48, 12\}$ .  $Pos_{a3}$  contains the information for the pick-up and delivery points of order three for the A supplies. AGV1 carries orders numbered  $\{9, 11, 3, \dots, 12\}$ , and by calculating the charging cost of each order, further charging scheduling is added based on the given initial power to complete the AGV scheduling plan as shown below. When the power of the AGV is insufficient to carry out the transportation work of the next order, a charging task will be inserted in time, which is represented by "0" in Figures 5 and 6. In order to prevent too many AGVs from going offline in the same period, we will check the overlapping period of the charging time window of AGVs and reconstruct the charging schedule using the large neighborhood search strategy.

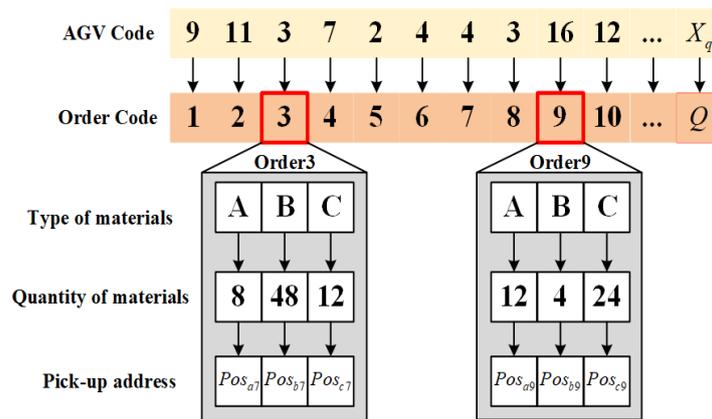


Figure 5. Schematic diagram of encoding and decoding.

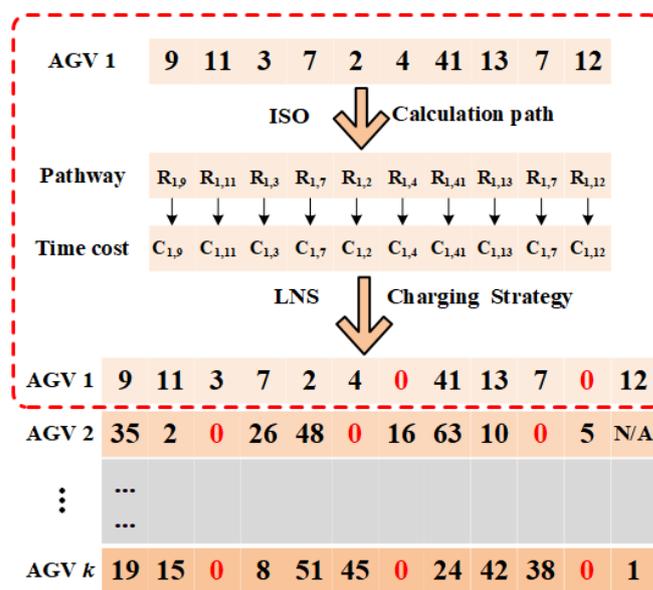


Figure 6. Schematic diagram of the AGV scheduling plan.

Figure 5 demonstrates the results of the three-level coding. For instance, the red box labeled 9 indicates that order 9 will be transported by AGV 16. Order 9 encapsulates details including material type, quantity, and pickup address. The algorithm can swiftly assign all orders to AGVs through efficient encoding and decoding techniques.

Figure 6 shows the charging plan inserted into the existing scheduling plan after calculating the electricity cost for each order. The red number 0 denotes the inserted charging plan, which specifies an available charging station and the time needed to fully recharge.

### 3.5. Large Neighborhood Search

Algorithm optimization efficiency is usually closely related to the operator settings, and some existing meta-heuristic algorithm operators lack fit with the real problem. This paper combines the task execution coding of each AGV in the multi-AGV cluster scheduling problem in the verification line and designs eight kinds of neighborhood operators to expand and optimize the adaptability of feasible solutions to achieve a more optimal local search effect. First, the feasible solution of one AGV is randomly selected as the benchmark information for the neighborhood search, and then the destruction and repair operations are executed sequentially to select the best one. The destruction and repair operations each contain four strategies: (1) delete a task on the rightmost side; (2) delete/add a task on the leftmost side; (3) randomly delete/add a task in the middle; and (4) randomly select one of the other feasible solutions and exchange one of the tasks.

Figure 7 illustrates the process of disrupting and repairing the operator encoding in the large neighborhood search algorithm. Specifically, part (a) shows the four cases where the task code is broken—the gray cells indicate removed tasks, while the yellow cells mark swapped tasks. Part (b) displays the process of repairing the operator encoding—the green cells represent new tasks, and the yellow cells indicate tasks that have been exchanged with others. This approach effectively disturbs the population structure, increases population diversity, and enhances the optimization capability of the algorithm.

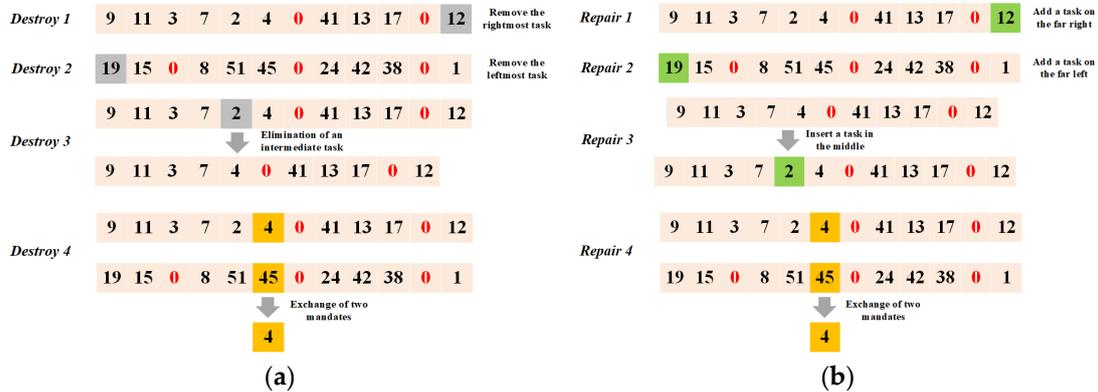


Figure 7. (a) Destructive operator; (b) restoration of the operator.

### 4. Simulation Experiment Results and Analysis

The algorithm programming tool in this paper is MATLAB R2021a, the operating system is Windows 11, the computer memory is 16G, and the CPU is Intel i7-12700H.

#### 4.1. Experiment 1: Algorithm Improvement Strategy Validation and Multi-Algorithm Data Comparison Analysis

To demonstrate the superiority of the ISO algorithm in terms of search efficiency, benchmark testing and comparison with particle swarm optimization (PSO), whale optimization algorithm (WOA), differential evolution (DE), grey wolf optimization (GWO), multiverse optimizer (MVO), genetic algorithm (GA), and basic SO were carried out. In order to comprehensively evaluate the performance of the proposed algorithm, 23 most commonly used standard test functions [35] are selected for simulation experiments and the parameter settings of each test function are shown in Table 4. The 23 benchmark data for the proposed algorithm are shown in Appendix A.

Table 4. Benchmark function.

Name	Function	$x_i$ Range	$F_{min}$
Sphere Function	$F_1(x) = \sum_{i=1}^{Dim} x_i^2$	$[-100,100]$	0
Schwefel's Problem 2.22	$F_2(x) = \sum_{i=1}^{Dim}  x_i  + \prod_{i=1}^{Dim}  x_i $	$[-10,10]$	0
Schwefel's Problem 1.2	$F_3(x) = \sum_{i=1}^{Dim} \left( \sum_{j=1}^i x_j \right)^2$	$[-100,100]$	0
Schwefel's Problem 2.21	$F_4(x) = \max_i \{ x_i , 1 \leq i \leq Dim\}$	$[-100,100]$	0
Generalized Rosenbrock's Function	$F_5(x) = \sum_{i=1}^{Dim} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30,30]$	0
Step Function	$F_6(x) = \sum_{i=1}^{Dim} ([x_i + 0.5])^2$	$[-100,100]$	0
Quartic Function i.e., Noise	$F_7(x) = \sum_{i=1}^{Dim} ix_i^4 + random[0,1)$	$[-1.28,1.28]$	0
Generalized Schwefel's Problem 2.26	$F_8(x) = \sum_{i=1}^{Dim} -x_i \sin(\sqrt{ x_i })$	$[-500,500]$	-12,569.5

Table 4. Cont.

Name	Function	$x_i$ Range	$F_{min}$
Generalized Rastrigin’s Function	$F_9(x) = \sum_{i=1}^{Dim} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]$	0
Ackley’s Function	$F_{10}(x) = -20 \exp(-0.2 \sqrt{\frac{1}{Dim} \sum_{i=1}^{Dim} x_i^2}) - \exp(\frac{1}{Dim} \sum_{i=1}^{Dim} \cos(2\pi x_i)) + 20 + e$	$[-32, 32]$	0
Generalized Griewank’s Function	$F_{11}(x) = \frac{1}{4000} \sum_{i=1}^{Dim} x_i^2 - \prod_{i=1}^{Dim} \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]$	0
Generalized Penalized Function	$F_{12}(x) = \frac{\pi}{30} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{29} (y_i - 1)^2\}$	$[-50, 50]$	0
Generalized Penalized Function	$F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{Dim} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_{Dim} - 1)^2 [1 + \sin^2(2\pi x_{Dim})] \} + \sum_{i=1}^{Dim} u(x_i, 5, 100, 4)$	$[-50, 50]$	0
Shekel’s Foxholes Function	$F_{14}(x) = \left( \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right)$	$[-65, 65]$	1
Kowalik’s Function	$F_{15}(x) = \sum_{i=1}^{11} \left[ a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	$[-5, 5]$	0.1484
Six-Hump Camel-Back Function	$F_{16} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]$	-1
Branin Function	$F_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	$[-5, 5]$	0.3
Goldstein-Price Function	$F_{18}(x) = \left[ 1 + (x_1 + x_2 + 1)^2 / 19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2 \right] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-2, 2]$	3
Hartman’s Family	$F_{19}(x) = - \sum_{i=1}^4 c_i \exp \left[ - \sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2 \right]$	$[1, 3]$	-3
	$F_{20}(x) = - \sum_{i=1}^4 c_i \exp \left[ - \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right]$	$[0, 1]$	-3
Shekel’s Family	$F_{21}(x) = - \sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]$	-1
	$F_{22}(x) = - \sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]$	-1
	$F_{23}(x) = - \sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	$[0, 10]$	-1

In order to fully verify the efficiency of the proposed algorithm in different dimensions, F1, F3, F4, F6, F7, F8, F10, F13, and F14 are selected as typical test functions to test the performance of each algorithm in the case of dimensions of 30 and 300, respectively. In this study, all algorithms underwent 100 iterations and each algorithm was executed 30 times. The experimental results in Tables 5 and 6 show that when the dimension is 30, the proposed algorithm is significantly better than other control algorithms in terms of convergence speed and solution accuracy, and all indicators are comprehensively improved. When the problem dimension is increased to 300, our algorithm still shows strong scalability despite the greatly increased complexity of the problem, and its performance indicators are still better than most control algorithms.

Table 5. Comparison results of the data obtained by running all 8 algorithms 30 times when the problem dimension is 30.

Algorithms	$F_1$	$F_3$	$F_6$	$F_7$	$F_8$	$F_{10}$	$F_{13}$	$F_{14}$
PSO	$2.31 \times 10^1$	$5.92 \times 10^2$	$2.67 \times 10^1$	$2.68 \times 10^{-1}$	$-8.76 \times 10^3$	$6.17 \times 10^0$	$3.24 \times 10^1$	$9.98 \times 10^{-1}$
WOA	$2.63 \times 10^{-34}$	$6.72 \times 10^3$	$4.79 \times 10^{-1}$	$2.40 \times 10^{-4}$	$-8.38 \times 10^3$	$8.88 \times 10^{-16}$	$3.16 \times 10^{-1}$	$9.98 \times 10^{-1}$
DE	$1.64 \times 10^1$	$2.47 \times 10^4$	$1.18 \times 10^1$	$1.16 \times 10^{-1}$	$-9.11 \times 10^3$	$2.44 \times 10^0$	$1.19 \times 10^0$	$9.98 \times 10^{-1}$
GWO	$1.37 \times 10^{-41}$	$5.28 \times 10^{-26}$	$2.87 \times 10^{-1}$	$1.39 \times 10^{-4}$	$-1.25 \times 10^4$	$8.88 \times 10^{-16}$	$7.75 \times 10^{-1}$	$9.98 \times 10^{-1}$
MVO	$1.82 \times 10^{-5}$	$2.86 \times 10^1$	$2.42 \times 10^{-1}$	$1.56 \times 10^{-3}$	$-1.25 \times 10^4$	$6.98 \times 10^{-16}$	$3.19 \times 10^{-1}$	$9.98 \times 10^{-1}$
GA	$2.51 \times 10^1$	$2.57 \times 10^3$	$1.63 \times 10^1$	$1.78 \times 10^{-1}$	$-1.25 \times 10^4$	$2.44 \times 10^{-13}$	$1.36 \times 10^0$	$9.98 \times 10^{-1}$
SO	$4.04 \times 10^{-37}$	$8.69 \times 10^{-25}$	$3.77 \times 10^{-2}$	$1.65 \times 10^{-4}$	$-1.26 \times 10^4$	$2.39 \times 10^{-13}$	$1.13 \times 10^{-5}$	$9.98 \times 10^{-1}$
ISO	$2.53 \times 10^{-56}$	$2.90 \times 10^{-43}$	$8.4 \times 10^{-3}$	$1.31 \times 10^{-5}$	$-1.26 \times 10^4$	$8.88 \times 10^{-16}$	$9.08 \times 10^{-3}$	$9.98 \times 10^{-1}$

Table 5. Cont.

Algorithms		$F_1$	$F_3$	$F_6$	$F_7$	$F_8$	$F_{10}$	$F_{13}$	$F_{14}$
Avg	PSO	$8.07 \times 10^1$	$2.25 \times 10^3$	$7.21 \times 10^1$	$5.26 \times 10^{-1}$	$-7.22 \times 10^3$	$8.34 \times 10^0$	$1.21 \times 10^4$	$4.62 \times 10^0$
	WOA	$3.44 \times 10^{-28}$	$2.49 \times 10^4$	$1.12 \times 10^0$	$7.78 \times 10^{-3}$	$-5.94 \times 10^3$	$2.30 \times 10^{-14}$	$1.02 \times 10^0$	$3.49 \times 10^0$
	DE	$2.47 \times 10^1$	$3.42 \times 10^4$	$2.30 \times 10^1$	$1.88 \times 10^{-1}$	$-8.51 \times 10^3$	$2.91 \times 10^0$	$3.88 \times 10^0$	$1.39 \times 10^0$
	GWO	$2.78 \times 10^{-39}$	$9.19 \times 10^{-21}$	$9.60 \times 10^{-1}$	$1.32 \times 10^{-3}$	$-1.04 \times 10^4$	$3.31 \times 10^0$	$1.58 \times 10^0$	$1.72 \times 10^0$
	MVO	$1.89 \times 10^{-8}$	$1.56 \times 10^{-13}$	$1.54 \times 10^1$	$1.33 \times 10^{-3}$	$-5.72 \times 10^3$	$2.62 \times 10^0$	$1.81 \times 10^0$	$2.93 \times 10^0$
	GA	$2.65 \times 10^1$	$3.98 \times 10^1$	$2.12 \times 10^0$	$2.09 \times 10^{-1}$	$-6.59 \times 10^4$	$1.39 \times 10^0$	$3.74 \times 10^1$	$1.66 \times 10^0$
	SO	$4.78 \times 10^{-34}$	$2.97 \times 10^{-20}$	$4.43 \times 10^0$	$6.45 \times 10^{-4}$	$-1.19 \times 10^4$	$2.80 \times 10^{-12}$	$1.58 \times 10^0$	$2.00 \times 10^0$
	ISO	$1.01 \times 10^{-5}$	$1.01 \times 10^{-35}$	$4.39 \times 10^{-2}$	$4.64 \times 10^{-4}$	$-1.21 \times 10^4$	$4.32 \times 10^{-15}$	$6.22 \times 10^{-2}$	$1.06 \times 10^0$
Std	PSO	$3.75 \times 10^1$	$1.06 \times 10^3$	$3.24 \times 10^1$	$1.74 \times 10^{-1}$	$7.99 \times 10^2$	$9.82 \times 10^{-1}$	$5.80 \times 10^4$	$4.27 \times 10^0$
	WOA	$8.65 \times 10^{-28}$	$9.58 \times 10^3$	$3.68 \times 10^{-1}$	$9.47 \times 10^{-3}$	$8.24 \times 10^2$	$1.20 \times 10^{-14}$	$3.18 \times 10^{-1}$	$3.32 \times 10^0$
	DE	$5.57 \times 10^0$	$4.76 \times 10^3$	$6.57 \times 10^0$	$3.56 \times 10^{-2}$	$3.40 \times 10^2$	$2.18 \times 10^{-1}$	$1.27 \times 10^0$	$9.56 \times 10^{-1}$
	GWO	$5.85 \times 10^{-39}$	$3.21 \times 10^{-20}$	$3.52 \times 10^{-1}$	$1.06 \times 10^{-3}$	$1.56 \times 10^3$	$7.52 \times 10^0$	$3.58 \times 10^{-1}$	$1.96 \times 10^0$
	MVO	$5.81 \times 10^{-12}$	$4.16 \times 10^1$	$3.49 \times 10^0$	$2.51 \times 10^{-2}$	$2.73 \times 10^3$	$2.85 \times 10^1$	$1.06 \times 10^0$	$8.05 \times 10^{-1}$
	GA	$3.61 \times 10^{-9}$	$3.68 \times 10^3$	$2.02 \times 10^1$	$3.09 \times 10^{-3}$	$1.98 \times 10^3$	$1.23 \times 10^{-7}$	$3.35 \times 10^1$	$3.54 \times 10^0$
	SO	$1.11 \times 10^{-33}$	$9.63 \times 10^{-20}$	$2.92 \times 10^0$	$5.01 \times 10^{-4}$	$8.15 \times 10^2$	$2.02 \times 10^{-12}$	$1.31 \times 10^0$	$1.89 \times 10^0$
	ISO	$1.44 \times 10^{-50}$	$4.03 \times 10^{-35}$	$4.34 \times 10^{-2}$	$2.77 \times 10^{-4}$	$7.92 \times 10^2$	$6.49 \times 10^{-16}$	$3.58 \times 10^{-2}$	$2.52 \times 10^{-1}$

Table 6. Comparison results of data from 8 algorithms run 30 times when the problem dimension is 300.

Algorithms		$F_1$	$F_3$	$F_4$	$F_6$	$F_7$	$F_8$	$F_{10}$	$F_{13}$
Best	PSO	$5.27 \times 10^4$	$3.76 \times 10^5$	$4.82 \times 10^1$	$4.85 \times 10^4$	$2.60 \times 10^2$	$-5.74 \times 10^4$	$1.67 \times 10^1$	$1.43 \times 10^8$
	WOA	$1.17 \times 10^{-30}$	$1.81 \times 10^6$	$7.13 \times 10^{-1}$	$1.52 \times 10^1$	$2.87 \times 10^{-4}$	$-6.34 \times 10^4$	$4.44 \times 10^{-15}$	$9.94 \times 10^0$
	DE	$2.65 \times 10^5$	$2.50 \times 10^6$	$9.78 \times 10^1$	$2.60 \times 10^5$	$2.98 \times 10^3$	$-2.85 \times 10^4$	$1.92 \times 10^1$	$3.11 \times 10^0$
	GWO	$4.67 \times 10^{-27}$	$8.51 \times 10^{-18}$	$3.45 \times 10^{-11}$	$6.37 \times 10^1$	$2.91 \times 10^{-4}$	$-1.00 \times 10^5$	$3.57 \times 10^{-11}$	$2.97 \times 10^1$
	MVO	$2.32 \times 10^{-5}$	$3.21 \times 10^5$	$5.26 \times 10^{-1}$	$1.62 \times 10^2$	$2.53 \times 10^{-3}$	$-1.25 \times 10^5$	$1.84 \times 10^{-6}$	$3.73 \times 10^6$
	GA	$1.96 \times 10^{-3}$	$2.16 \times 10^3$	$6.94 \times 10^1$	$2.75 \times 10^3$	$1.75 \times 10^3$	$-2.56 \times 10^4$	$1.59 \times 10^1$	$2.52 \times 10^1$
	SO	$3.73 \times 10^{-24}$	$1.76 \times 10^{-16}$	$2.39 \times 10^{-12}$	$6.21 \times 10^{-1}$	$8.80 \times 10^{-5}$	$-1.26 \times 10^5$	$1.21 \times 10^{-11}$	$7.22 \times 10^{-1}$
	ISO	$3.10 \times 10^{-32}$	$3.55 \times 10^{-20}$	$1.94 \times 10^{-12}$	$6.26 \times 10^{-1}$	$1.14 \times 10^{-4}$	$-1.26 \times 10^5$	$4.44 \times 10^{-15}$	$1.57 \times 10^{-2}$
Avg	PSO	$6.46 \times 10^4$	$6.58 \times 10^5$	$5.50 \times 10^1$	$6.75 \times 10^4$	$4.39 \times 10^2$	$-5.22 \times 10^4$	$1.77 \times 10^1$	$4.11 \times 10^8$
	WOA	$1.54 \times 10^{-26}$	$3.09 \times 10^6$	$4.79 \times 10^1$	$2.75 \times 10^1$	$1.28 \times 10^{-2}$	$-4.16 \times 10^4$	$4.41 \times 10^{-14}$	$1.70 \times 10^1$
	DE	$2.85 \times 10^5$	$3.47 \times 10^6$	$9.85 \times 10^1$	$2.88 \times 10^5$	$3.91 \times 10^3$	$-2.70 \times 10^4$	$1.95 \times 10^1$	$4.16 \times 10^0$
	GWO	$1.90 \times 10^{-23}$	$4.44 \times 10^{-11}$	$5.43 \times 10^{-11}$	$6.78 \times 10^1$	$2.28 \times 10^{-3}$	$-8.52 \times 10^4$	$5.60 \times 10^0$	$2.99 \times 10^1$
	MVO	$3.95 \times 10^{-6}$	$3.13 \times 10^5$	$5.02 \times 10^{-1}$	$2.51 \times 10^2$	$1.99 \times 10^{-3}$	$-2.23 \times 10^4$	$2.63 \times 10^{-3}$	$1.62 \times 10^2$
	GA	$2.05 \times 10^1$	$3.49 \times 10^6$	$2.81 \times 10^1$	$1.95 \times 10^1$	$1.67 \times 10^2$	$-2.89 \times 10^4$	$1.84 \times 10^{-2}$	$2.35 \times 10^5$
	SO	$2.01 \times 10^{-21}$	$2.33 \times 10^{-10}$	$1.12 \times 10^{-11}$	$5.07 \times 10^1$	$5.40 \times 10^{-4}$	$-1.22 \times 10^5$	$1.60 \times 10^{-10}$	$1.20 \times 10^1$
	ISO	$4.33 \times 10^{-30}$	$8.31 \times 10^{-11}$	$1.31 \times 10^{-11}$	$1.84 \times 10^0$	$5.12 \times 10^{-4}$	$-1.23 \times 10^5$	$4.56 \times 10^{-15}$	$1.36 \times 10^0$
Std	PSO	$7.17 \times 10^3$	$1.95 \times 10^5$	$4.56 \times 10^0$	$8.96 \times 10^3$	$1.05 \times 10^2$	$3.31 \times 10^3$	$4.39 \times 10^{-1}$	$1.44 \times 10^8$
	WOA	$4.60 \times 10^{-26}$	$1.01 \times 10^6$	$2.20 \times 10^1$	$6.00 \times 10^0$	$1.86 \times 10^{-2}$	$8.63 \times 10^3$	$9.74 \times 10^{-14}$	$4.36 \times 10^0$
	DE	$1.12 \times 10^4$	$3.95 \times 10^5$	$2.91 \times 10^{-1}$	$1.20 \times 10^4$	$3.15 \times 10^2$	$7.53 \times 10^2$	$1.54 \times 10^{-1}$	$4.67 \times 10^8$
	GWO	$6.48 \times 10^{-23}$	$1.95 \times 10^{-10}$	$1.25 \times 10^{-10}$	$1.35 \times 10^0$	$1.40 \times 10^{-3}$	$7.05 \times 10^3$	$8.35 \times 10^0$	$6.66 \times 10^{-2}$
	MVO	$1.76 \times 10^{-2}$	$1.75 \times 10^{-4}$	$2.11 \times 10^{-5}$	$1.85 \times 10^2$	$1.19 \times 10^{-2}$	$5.86 \times 10^3$	$1.43 \times 10^{-4}$	$2.52 \times 10^6$
	GA	$1.34 \times 10^3$	$2.59 \times 10^2$	$1.62 \times 10^1$	$1.26 \times 10^3$	$2.05 \times 10^3$	$4.47 \times 10^3$	$1.37 \times 10^{-1}$	$1.63 \times 10^2$
	SO	$3.13 \times 10^{-21}$	$1.14 \times 10^{-9}$	$6.63 \times 10^{-12}$	$2.95 \times 10^1$	$5.12 \times 10^{-4}$	$5.85 \times 10^3$	$7.93 \times 10^{-11}$	$1.27 \times 10^1$
	ISO	$7.30 \times 10^{-30}$	$4.54 \times 10^{-10}$	$8.88 \times 10^{-12}$	$7.85 \times 10^{-1}$	$3.39 \times 10^{-4}$	$4.46 \times 10^3$	$6.49 \times 10^{-16}$	$7.65 \times 10^{-1}$

4.2. Experiment 2: Solving Multi-AGV Scheduling Model of Automated Meter Verification Workshop Taking into Account Charging Constraints

This study constructs a simulation environment with 10 AGVs and 300 transport tasks. The initialization information of each AGV includes the initial battery level, initial position, and maximum payload capacity. Table 7 contains basic information such as the type and size of the meter. Each transport task or order contains a storage point coordinate (X,Y) in the grid map in Figure 2, a delivery point coordinate, cargo type, quantity, and weight. Table 8 shows the order data used in the experiment, which contains the meter type, quantity, weight, storage location and delivery point for 300 transport tasks. For example, Storage coordinates (48,3) indicates that the storage point is located in X = 48 and Y = 3 in the raster map.

Table 7. Metering material information.

Cargo Type	Name	Crate Specifications	Dimensions (L × W × H)	Maximum Load Per Box
A	single-phase energy meter	12 pcs/box	720 × 450 × 120 (mm)	25 kg/box
B	three-phase energy meter	4 pcs/box		
C	Low Voltage Transformers	12 pcs/box	720 × 450 × 200 (mm)	50 kg/box

**Table 8.** Initializing Orders.

Order Number	1	2	3	4	5	6	7	8	9	...	299	300
Storage coordinates(X,Y)	48,3	34,3	41,3	47,13	54,4	54,17	41,10	33,4	33,5	...	54,21	33,10
Delivery point(X,Y)	4,21	8,21	16,21	16,21	8,21	4,21	12,21	12,21	8,21	...	12,21	4,21
Meter type	C	C	A	C	B	A	B	B	B	...	C	B
Weight of material	48	42	45	12	24	45	27	15	54	...	15	27
Quantity of material	24	21	15	6	12	21	15	9	27	...	9	15

The goal of the simulation is to optimally schedule and route the AGVs to complete the 300 transport tasks under constraints such as battery life and payload capacity. This simulated environment allows flexible initialization of AGVs and transport tasks to evaluate different AGV scheduling and routing algorithms. Key performance indicators, including total travel distance, charging cost, etc., can be analyzed through the simulation. This simulation environment enables rapid debugging and comparison of scheduling methods before actual implementation to identify the optimal approach. Table 9 shows the initial information of the AGV, including the initial position, maximum weight limit and initial battery level.

**Table 9.** Initializing AGV parameters.

	AGV1	AGV2	AGV3	AGV4	AGV5	AGV6	AGV7	AGV8	AGV9	AGV10
Initial charge	96%	97%	82%	75%	95%	100%	99%	90%	98%	93%
Initial position(X,Y)	10,7	15,5	11,19	40,8	41,9	55,13	14,12	33,15	47,22	30,1
Maximum load (kg)	120	120	120	120	120	120	120	120	120	120

Due to the high time and space complexity of the gray wolf algorithm, the GWO algorithm is not used in Experiment 2, and it is worth noting that this does not affect the experimental results. The parameters of the PSO algorithm include acceleration factor  $c1$ , denoted as 1.5, acceleration factor  $c2$ , denoted as 1.9, number of particle populations, denoted as 30, and inertia weight,  $w$ , denoted as 0.7. The GA algorithm has its crossover probability set to 0.7, mutation probability set to 0.1, and two individuals with the best fitness are retained per generation. The probability of the predation mechanism of the whale algorithm is set to 0.5. The population size of the DE algorithm is set to 30, the mutation probability is set to 0.5, and the crossover probability is set to 0.9. The travel distance rate parameter  $H$  of the MVO algorithm is set to 0.5. The food amount threshold  $Q$  of the SO algorithm is set to 0.25, the environment temperature  $Temp$  is set to 0.6, and the random number threshold for entering battle or mating modes is set to 0.6. The population size of the ISO algorithm is set to  $Noop = 30$ , and the maximum number of iterations is  $G = 100$ . The average objective function curves of each of the seven algorithms over 30 runs are shown in Figure 8.

The iteration plots are obtained by taking an average of 30 independent runs to examine the performance of the proposed algorithm. The simulation results show that for this AGV fleet scheduling optimization problem in the automated electric meter verification workshops, ISO exhibits better convergence and finds the global optimal solution faster and more reliably than other algorithms. This is mainly attributed to the introduced random reverse learning mechanism that avoids getting trapped in local optima, and the Gaussian mutation operator that maintains population diversity, striking a better balance between global and local search capabilities.

Based on the aforementioned simulation results, this study further optimizes the charging of AGVs. The charging optimization strategy considers the remaining battery level, travel distance, and distance to charging stations of each AGV to rationally schedule the charging time and station selection. By combining the global optimization advantages of large neighborhood search to replan the charging tasks of the AGVs, the charging frequency and cost of each AGV are recorded. Comparative experiments with other algorithms

demonstrate that the proposed charging optimization strategy significantly reduces the charging frequency and cost of the AGVs. This is mainly attributed to the strategy’s accurate estimation of remaining battery life and reasonable scheduling of charging time, along with the global optimization of the charging plan enabled by a large neighborhood search.

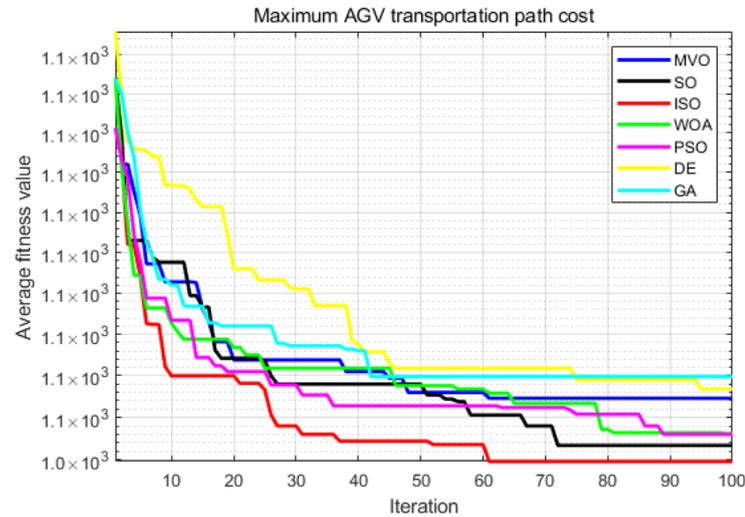


Figure 8. Iteration curves for 30 algorithm runs.

The traditional material scheduling in automated electric meter verification workshops adopts the FIFO principle, assigning orders sequentially to AGVs based on order arrival times. Table 10 compares the traditional FIFO method with various algorithms, showing that the proposed ISO algorithm can significantly reduce the number of chargings and charging costs. The sum of all costs is the objective function value.

Table 10. Comparison of results from each algorithm.

Algorithms Number of AGVs	FIFO		PSO		GA		MVO		DE		WOA		SO		ISO	
	Times	Costs														
AGV1	9	1955	7	1538	6	1700	8	1728	8	1721	7	1734	7	1679	6	1566
AGV2	9	1887	7	1764	7	1757	7	1692	7	1701	7	1592	7	1715	6	1508
AGV3	9	1865	8	1719	8	1728	8	1567	8	1730	7	1710	6	1689	5	1585
AGV4	8	1787	7	1646	6	1741	6	1676	6	1635	6	1681	6	1674	6	1602
AGV5	8	1741	7	1689	8	1761	6	1708	6	1654	6	1735	6	1682	5	1523
AGV6	9	1735	8	1659	7	1627	6	1764	6	1761	6	1710	6	1704	5	1482
AGV7	8	1709	7	1594	8	1689	8	1675	7	1736	8	1750	7	1682	5	1518
AGV8	9	1763	8	1654	7	1697	7	1668	7	1561	7	1667	7	1681	5	1493
AGV9	8	1720	8	1713	7	1573	7	1752	7	1764	7	1749	7	1569	5	1583
AGV10	8	1715	7	1712	8	1534	8	1748	7	1658	7	1753	7	1717	5	1490
Totals	85	17,877	74	16,688	72	16,807	71	16,978	69	16,921	68	17,081	66	16,792	53	15,350

In order to compare the performance of different optimization algorithms for AGV charging route planning, this study analyzed the simulation results of the FIFO, MVO, WOA, DE, GA, PSO, SO, and ISO algorithms. The charging times of these algorithms are 85, 74, 72, 71, 69, 68, 66, and 53, and the costs are 17,877, 16,688, 16,807, 16,978, 16,921, 17,081, 16,792, 15,350, respectively. It is worth noting that ISO has the largest improvement over the traditional FIFO method, reducing costs by 16.46% while reducing the number of charges by 60%. In addition, compared with similar algorithms, ISO reduces the cost by 8.71%, 9.49%, 10.60%, 10.23%, 11.27%, and 9.39%, respectively. In terms of charging times, ISO was reduced by 39.62%, 35.84%, 33.96%, 30.18%, 28.30%, and 24.52%, respectively. The results show that the large neighborhood search strategy can effectively minimize the cost and greatly reduce the number of charging times, which further verifies the effectiveness of the charging strategy.

#### 4.3. Algorithm Time Complexity Analysis

In the basic snake optimization algorithm, the exploration mode is equally divided into two populations, thus the time complexity is  $O(2 * Noop^2)$ . In the battle mode, the male population completes the position update through a cyclic operation, and the battle factor of each dimension is calculated at the same time, thus the time complexity is  $O(2 * Noop^{2*dim})$  and the time complexity of the female mating mode is expressed as  $O(2 * Noop^{2*dim})$ . Therefore, the overall time complexity of the standard snake optimization algorithm is  $O(G * \max(O(2 * Noop^2), O(2 * Noop^{2*dim}))) = O(2G * Noop^{2*dim})$ . The ROBL strategy is applied to the initialization phase of the population, which has a time complexity of  $O(1)$ . In the Individualized memory strategy, the male population is updated with partial information from the eliminated individuals through a round-robin operation, and the time complexity of the operation is expressed as  $O(4 * Noop^{2*dim})$ . The Gaussian variational strategy targets the worst individuals in the female population and therefore does not manipulate Gaussian variation through cycling.

Therefore, the overall time complexity of the ISO algorithm is as follows:

$$O(G * \max(O(1), O(2 * Noop^2), O(2 * Noop^{2*dim}), O(4 * Noop^{2*dim}))) = O(4G * Noop^{2*dim})$$

In summary, although the time complexity of the ISO algorithm  $O(4G * Noop^{2*dim})$  is slightly higher than that of the SO algorithm  $O(2G * Noop^{2*dim})$ , the overall time complexity of the ISO algorithm shows advantages in the test and simulation experiments of the benchmark function, especially in terms of algorithm convergence, global search speed and global search accuracy. Therefore, the time complexity of the ISO algorithm is considered acceptable.

#### 4.4. Parameter Sensitivity Analysis

In the logistics environment of the automated meter verification workshops, the size of the AGV fleet and the number of tasks are uncertain. As the task volume increases, a smaller number of AGVs will cause the material to not be delivered within the time window. Too many AGVs can make the environment too complex, and a lot of additional paths and costs need to be added to avoid collisions and other situations. Therefore, determining the matching relationship between the size of the AGV fleet and the change in order volume is key to improving the efficiency of AGV transportation and ensuring timely delivery.

To further validate the reliability of the proposed algorithm, this study conducted comparative experiments on scales of 100, 300, and 500 transportation tasks, and 5 and 10 AGV quantities. Figures 9 and 10 show that as the scale increases, the advantage of the proposed algorithm over other algorithms becomes more significant, with a remarkable reduction in the charging frequency of the AGVs. This demonstrates that for larger-scale AGV scheduling and charging optimization problems, the algorithm maintains strong global search capabilities and avoids getting trapped in local optima, thus achieving better optimization performance. As the scale continues to expand, the designed algorithm exhibits good scalability and stability.

In order to comprehensively evaluate the solution performance of the proposed algorithm under different parameter settings, we designed multiple sets of comparative experiments based on two factors: population size and number of iterations. First, the population size was set to 10, 30, and 50, and the number of iterations was 50, 100, and 200, respectively, and the influence of the number of iterations and population size on the solution efficiency of the algorithm was evaluated. As can be seen in Figure 11, as the number of iterations increases, the feasible solution approximates the optimal value; however, the computational time cost also increases linearly. On the other hand, because the algorithm adopts a population design that is divided into male and female groups, the small population size cannot give full play to the advantages brought by population diversity. The large population size leads to an increase in time cost of more than 50% when a similarly accurate solution is obtained. Combining the indicators, the combination of

parameters with 100 iterations and a population size of 30 allows the algorithm to achieve the best balance, which has obvious comprehensive advantages for solving the model.

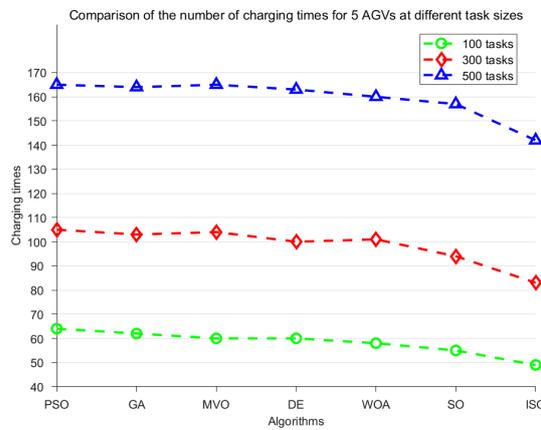


Figure 9. Comparison of the number of charging times for 5 AGVs with different task sizes.

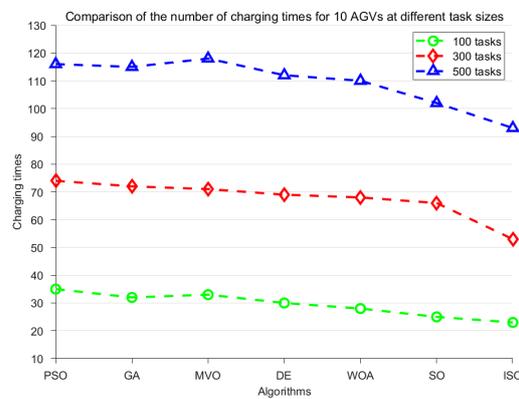


Figure 10. Comparison of the number of charging times for 10 AGVs with different task sizes.

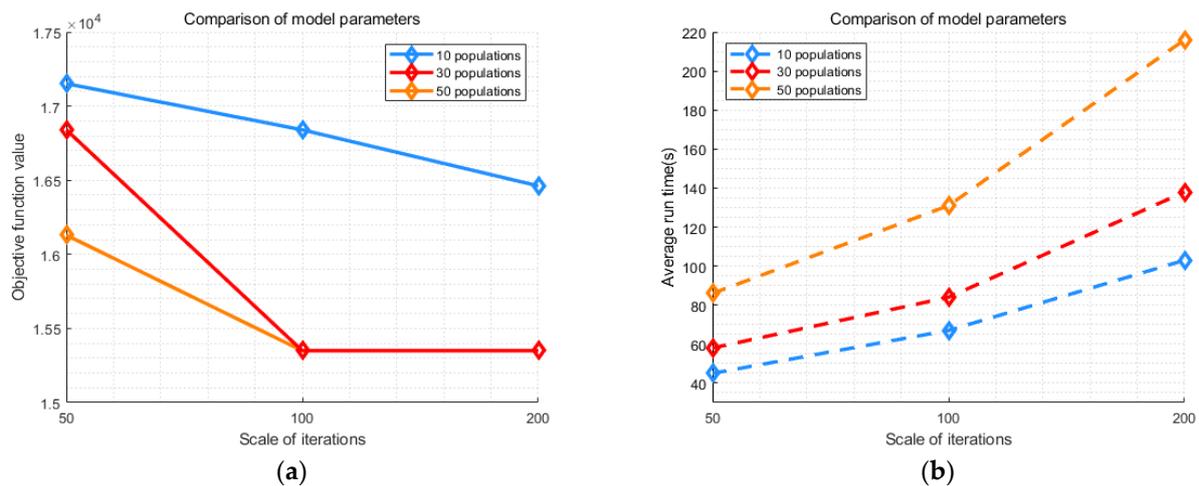


Figure 11. (a) The effect of model parameters on the value of the objective function; (b) the effect of model parameters on the running time of the algorithm.

### 5. Results

In automated electric meter verification workshops, different verification requirements for various electric meters can be met by dynamically matching AGVs with orders. Table 11 below shows the matching results of 10 AGVs and 300 orders, where each AGV is assigned a set of orders (see Table 11). By dynamically combining AGVs and orders, this method

can automate the verification operations for different types of electric meters on demand, providing an effective solution for production optimization in the workshop. The flexible assignment of AGVs to diverse orders enables customized verification for a range of meter specifications within the same automated intelligent workshop. Table 11 shows all AGVs and orders shipped for 100 iterations and 30 populations solved by the ISO algorithm. The order in which orders are fulfilled is optimized to achieve the lowest possible distribution costs. When the total weight and total volume of the order are just not more than the limit of the AGV, it is deemed that the AGV has completed a transportation task, and “[.]” in Table 11 One task for AGV. [186,92] indicates that the first task of AGV1 is order 186 and order 96. AGV1 first arrives at all pick-up points of order 186 from the initial location. After a fixed pick-up time window, the goods for order 186 are fully loaded; the AGV then goes to the pick-up point for order 92 from the current location, and when a transportation task is completed, it goes to the material transfer point to complete the delivery. To avoid confusion going forward, let’s refer to the first task assigned to AGV1 as “Path1”. The AGV will not exceed the load during a single mission.

**Table 11.** AGV task scheduling information.

AGV Number	Carrier Orders
AGV1	[186,92]→[131,168,109,253,225]→[114,72,41]→[22,216,144]→[100,90,107]→[286,95,247,138]→[258,50,60,178]→[111,148,273]→[194,271,12]
AGV2	[5,197,284,85]→[165,48,30]→ [192,80,214,264]→[183,294]→[38,279]→[25,10]→[161,98,282]→[52,207]→[202,238,222,137]→[28,120,134,21]
AGV3	[55,64,267,201,20,54]→[295,23]→[37,62,276]→[3,119,176,260]→[24,36,146,280]→[93,46,118,254]→[8,126,205,160,223]→[58,163]
AGV4	[167,252,70,191]→[269,43]→[1,33,78,113,239]→[200,256,71]→[208,162,136]→[177,249]→[175,232,115]→[129,219]→[149,6,240]→[45,39,94]
AGV5	[66,166,204,49]→[227,159,300,103]→[19,56,106,59]→[112,91,82]→[110,128]→[147,265,241]→[242,281,121,188]→[261,196,278]→[140,170]→245
AGV6	[174,13,226]→[198,203,274]→[105,248]→[180,32,75,234]→[209,150]→[132,51]→[157,206,74]→[97,285,145,236,154]→[139,185,77,123]→[11,73]
AGV7	[231,47,151]→[298,9]→[35,268,244,277]→[27,164,101]→[292,117,124]→[259,67,217,228,288]→[270,96,290,34]→[42,116,230]→[53,7,29]
AGV8	[69,141,142]→[89,44,86]→[135,133,173]→[297,211,88]→[218,283]→[16,187]→[108,246,262,63,190]→[84,153,130,143]→[61,266]→[224,210,272]
AGV9	[181,182]→[195,15]→[220,289]→[40,152,221]→[250,235]→[251,243,122]→[102,81,26]→[199,2,237]→[287,215,171]→[169,83,18]→[155,299,257]→76
AGV10	[193,4,87,125]→[158,99]→[156,14,104,127]→[291,31]→[172,296,275]→[57,255]→[263,229,179]→[65,184]→[293,68,17]→[213,189,233,79]→212

Figures 12–16 show the routes for some AGVs, the red five-pointed star indicates the location of AGV1. The red line depicts AGV1’s route, while the green coordinate points along the path are the orders carried by AGV1. Specific details on these orders are provided in Table 11. The results demonstrate that the planned routes align with the actual transportation demands of the orders. Taking the transportation and charging trajectory of AGV1 as an example, this study visually validated the effectiveness of the obtained optimization scheme and transportation plan by mapping its complete transportation and charging routes. AGV1’s transportation and charging trajectory covered multiple transport lines in the workshop, fully considering factors like transport distance, remaining battery level, and charging station distribution, and rationally arranging the timing and location of charging. This example demonstrates that the designed charging optimization strategy is capable of scientifically and rationally guiding transportation scheduling and charging management of AGVs.



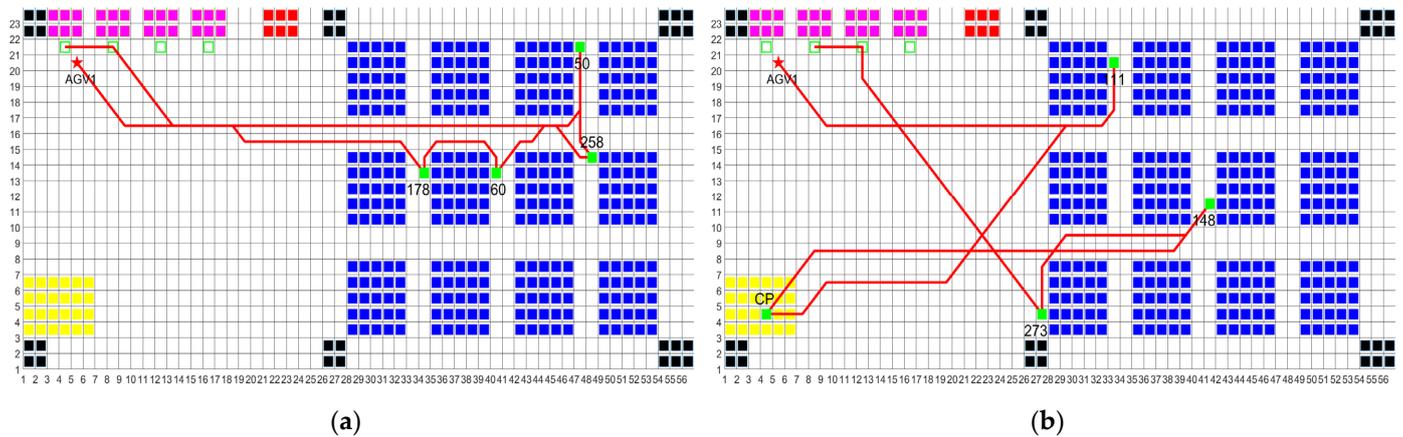


Figure 15. (a) Path 7 route for AGV1; (b) Path 8 route for AGV1.

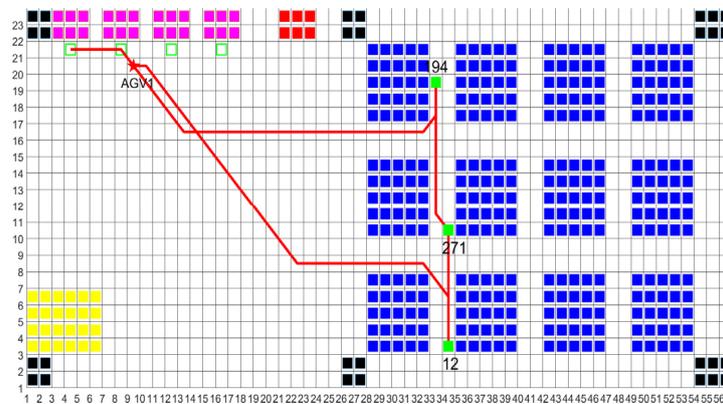


Figure 16. Path 9 route for AGV1.

In Figure 17, it can be clearly observed that when the transportation routes for AGV1 and AGV2 intersect, both trajectories exhibit significant offsets, successfully avoiding collision. Meanwhile, the comparative analysis of the coordinate data for AGV1 and AGV2 at each timestamp in Table 12 further verifies that a safe distance is consistently maintained between the two vehicles without trajectories overlapping at any given moment. It is worth noting that the decision-making characteristics of the proposed obstacle avoidance strategy are shown in the area of shelf and material transfer points, the obstacles in the shelf area are distributed in a rectangular shape, and the frequent turning of AGVs leads to the conflict between AGVs, mainly in non-head-on collisions. The four material transfer points are linearly distributed, and although there are fewer AGVs turning, due to the small area of material transfer points, it is easy to gather more AGVs, which leads to more head-on collisions. In Figure 17, it can be observed that the routes for AGVs in the corner area of the shelf overlap greatly because, in the event of a non-head-on collision, the AGVs choose to wait instead of detour according to the assigned priority of way. In contrast, two distinctly staggered paths can be seen in the material transfer point area, as AGVs choose to avoid them in the event of a contralateral conflict. The data in Table 12 fully verify the effectiveness of the proposed obstacle avoidance strategy. The above analysis fully proves that the designed collision avoidance strategy can effectively guide AGVs to safely maneuver around obstacles in complex environments and prevent collisions between vehicles, providing strong support for intelligent and collaborative transportation in workshops.

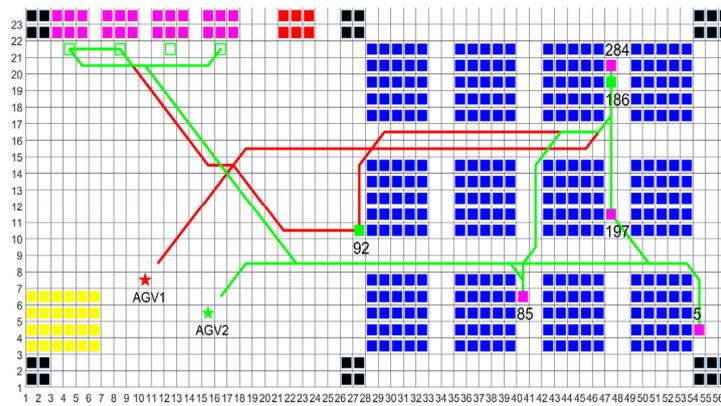


Figure 17. AGV1 and AGV2 collaborative path planning.

Table 12. AGV path coordinates and moment-to-moment information.

AGV1			AGV2		
Path Coordinates (X,Y)	Timestamp		Path Coordinates (X,Y)	Timestamp	
Ending coordinates			Ending coordinates		
...	...		...	...	
35	15	(288)	38	8	(288)
34	15	(287)	39	8	(287)
33	15	(286)	40	7	(286)
32	15	(285)	40	6	(285)
...	...	...	...	...	...
54	13	(138)	8	21	(138)
54	14	(137)	7	21	(137)
53	15	(136)	6	21	(136)
52	16	(135)	5	21	(135)
51	16	(134)	4	21	(134)
...	...	...	...	...	...
20	15	(10)	25	3	(10)
19	15	(9)	24	3	(9)
18	15	(8)	23	3	(8)
17	14	(7)	22	3	(7)
16	13	(6)	21	3	(6)
15	12	(5)	20	3	(5)
14	11	(4)	19	3	(4)
13	10	(3)	18	3	(3)
12	9	(2)	17	3	(2)
11	8	(1)	16	4	(1)
starting coordinates		(0)	starting coordinates		(0)

### 6. Conclusions

In view of the complex environment of automated electric meter verification workshops, we considered multi-dimensional factors such as AGV's path planning, collaborative obstacle avoidance, and charging constraints, and constructed a multi-AGV scheduling model of automated meter verification workshop, taking into account charging constraints, with the goal of minimizing distribution cost and charging cost. The innovation of the model lies in the multi-stage optimization of the electric meter transportation task, charging task, and path planning, and the total cost of transportation cost and charging task is minimized on the basis of obstacle avoidance. In order to solve the model, we propose a snake optimization algorithm that integrates the improvement of individual memory update strategy and Gaussian variational strategy in which the Gaussian variational strategy perturbs poorly located individuals through a certain probability, which effectively enhances the global search ability of the algorithm, and the individual memory update

strategy makes full use of the historical search information of individuals and effectively enhances the local search ability of the algorithm.

First, the order is allocated to the AGV through ISO to achieve the shortest distribution path; second, a large neighborhood search strategy is proposed to achieve the minimum charging cost for AGVs to solve the problem of charging task scheduling; finally, for the problem of AGV path planning and obstacle avoidance, we consider obstacle avoidance between static obstacles in the environment and dynamic AGVs at the same time, and propose an improved A\* algorithm to realize a precise obstacle avoidance collaborative AGV operation. In the simulation experiments, the benchmark test, case analysis, and parameter sensitivity analysis of the algorithm verify the effectiveness of the proposed algorithm in solving high-dimensional problems, such as path cost and reducing the number of charging times.

However, there are still limitations to the current research process. For example, the time complexity of ISO is slightly higher than that of some common algorithms, and the solution efficiency of the algorithm needs to be further improved to adapt to different application scenarios. In addition, the congestion of multiple AGVs is ignored in the path planning and too much congestion will lead to too much time being wasted on obstacle avoidance. In future research, we will further improve the efficiency of the algorithm and combine more practical application requirements to ensure that the AGV scheduling problem in complex environments can be effectively solved.

**Author Contributions:** Conceptualization, K.S.; methodology, K.S. and M.Z.; software, N.P. and Z.H.; validation, S.Y., Z.A. and N.P.; formal analysis, K.S. and M.Z.; resources, S.Y. and N.P.; data curation, N.P.; writing—original draft preparation, Z.A.; writing—review and editing, Z.A.; visualization, K.S. and M.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the Science and technology project of China Southern Power Grid Co., Ltd., (YNKJXM20220174).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Due to the nature of this research, participants in this study did not agree for their data to be shared publicly, thus supporting data is not available.

**Conflicts of Interest:** The authors have no conflict of interest to declare.

## Appendix A Test Results for the Algorithms

**Table A1.** PSO, WOA, DE, MVO, GA, SO and ISO test results in experiment 1.

Fitness		PSO	WOA	DE	MVO	GA	SO	ISO
$F_1$	Mean	$7.68 \times 10^1$	$3.51 \times 10^{-26}$	$2.11 \times 10^1$	$3.88 \times 10^4$	$1.04 \times 10^{-33}$	$1.30 \times 10^{-39}$	$2.58 \times 10^{-50}$
	Std	$4.07 \times 10^1$	$1.81 \times 10^{-25}$	$6.05 \times 10^0$	$7.06 \times 10^3$	$2.56 \times 10^{-33}$	$2.97 \times 10^{-39}$	$5.92 \times 10^{-50}$
$F_2$	Mean	$2.01 \times 10^1$	$1.82 \times 10^{-19}$	$1.16 \times 10^0$	$9.62 \times 10^{15}$	$1.98 \times 10^{-12}$	$1.22 \times 10^{-19}$	$2.50 \times 10^{-27}$
	Std	$5.29 \times 10^0$	$5.91 \times 10^{-19}$	$1.49 \times 10^{-1}$	$2.79 \times 10^{16}$	$1.48 \times 10^{-12}$	$3.02 \times 10^{-19}$	$4.57 \times 10^{-27}$
$F_3$	Mean	$2.27 \times 10^3$	$2.75 \times 10^4$	$3.44 \times 10^4$	$4.47 \times 10^5$	$1.77 \times 10^{-20}$	$1.41 \times 10^{-21}$	$1.59 \times 10^{-37}$
	Std	$1.01 \times 10^3$	$9.48 \times 10^3$	$5.01 \times 10^3$	$4.79 \times 10^5$	$5.71 \times 10^{-20}$	$3.45 \times 10^{-21}$	$4.52 \times 10^{-37}$
$F_4$	Mean	$2.02 \times 10^1$	$3.15 \times 10^1$	$4.03 \times 10^1$	$6.90 \times 10^1$	$3.64 \times 10^{-13}$	$1.98 \times 10^{-13}$	$7.57 \times 10^{-23}$
	Std	$5.88 \times 10^0$	$1.72 \times 10^1$	$3.47 \times 10^0$	$5.07 \times 10^0$	$6.19 \times 10^{-13}$	$2.47 \times 10^{-13}$	$1.07 \times 10^{-22}$
$F_5$	Mean	$8.22 \times 10^4$	$2.87 \times 10^1$	$2.24 \times 10^3$	$4.01 \times 10^7$	$2.47 \times 10^1$	$2.47 \times 10^1$	$2.71 \times 10^1$
	Std	$7.22 \times 10^4$	$1.68 \times 10^{-1}$	$6.51 \times 10^2$	$1.68 \times 10^7$	$9.17 \times 10^0$	$9.14 \times 10^0$	$9.34 \times 10^{-1}$
$F_6$	Mean	$7.95 \times 10^1$	$1.24 \times 10^0$	$2.12 \times 10^1$	$4.14 \times 10^4$	$4.87 \times 10^0$	$5.42 \times 10^{-2}$	$9.72 \times 10^{-1}$
	Std	$3.53 \times 10^1$	$3.98 \times 10^{-1}$	$5.89 \times 10^0$	$8.83 \times 10^3$	$2.67 \times 10^0$	$3.42 \times 10^{-2}$	$4.07 \times 10^{-1}$

Table A1. Cont.

Fitness		PSO	WOA	DE	MVO	GA	SO	ISO
F <sub>7</sub>	Mean	$4.48 \times 10^{-1}$	$7.68 \times 10^{-3}$	$1.77 \times 10^{-1}$	$1.95 \times 10^{-1}$	$6.34 \times 10^{-4}$	$4.82 \times 10^{-4}$	$9.93 \times 10^{-4}$
	Std	$1.66 \times 10^{-1}$	$9.10 \times 10^{-3}$	$3.50 \times 10^{-2}$	$6.23 \times 10^{-2}$	$6.37 \times 10^{-4}$	$2.78 \times 10^{-4}$	$1.01 \times 10^{-3}$
F <sub>8</sub>	Mean	$-7.08 \times 10^3$	$-5.76 \times 10^3$	$-8.48 \times 10^3$	$-6.45 \times 10^3$	$-1.23 \times 10^4$	$-1.19 \times 10^4$	$-9.84 \times 10^3$
	Std	$6.78 \times 10^2$	$5.86 \times 10^2$	$3.73 \times 10^2$	$7.89 \times 10^2$	$5.48 \times 10^2$	$8.64 \times 10^2$	$1.46 \times 10^3$
F <sub>9</sub>	Mean	$1.06 \times 10^2$	$1.93 \times 10^{-1}$	$9.98 \times 10^1$	$1.36 \times 10^2$	$1.82 \times 10^1$	$1.99 \times 10^1$	$0.00 \times 10^0$
	Std	$1.67 \times 10^1$	$1.06 \times 10^0$	$7.44 \times 10^0$	$2.68 \times 10^1$	$2.03 \times 10^1$	$2.49 \times 10^1$	$0.00 \times 10^0$
F <sub>10</sub>	Mean	$8.14 \times 10^0$	$1.92 \times 10^{-14}$	$2.89 \times 10^0$	$1.82 \times 10^1$	$2.37 \times 10^{-12}$	$4.44 \times 10^{-15}$	$6.46 \times 10^{-1}$
	Std	$9.78 \times 10^{-1}$	$1.10 \times 10^{-14}$	$2.27 \times 10^{-1}$	$5.40 \times 10^{-1}$	$1.61 \times -12$	$0.00 \times 10^0$	$3.54 \times 10^0$
F <sub>11</sub>	Mean	$1.02 \times 10^0$	$1.33 \times 10^{-2}$	$1.21 \times 10^0$	$4.06 \times 10^2$	$0.00 \times 10^0$	$0.00 \times 10^0$	$0.00 \times 10^0$
	Std	$3.38 \times 10^{-2}$	$7.30 \times 10^{-2}$	$6.11 \times 10^{-2}$	$5.80 \times 10^1$	$0.00 \times 10^0$	$0.00 \times 10^0$	$0.00 \times 10^0$
F <sub>12</sub>	Mean	$2.61 \times 10^1$	$9.51 \times 10^{-2}$	$1.21 \times 10^0$	$1.22 \times 10^8$	$5.50 \times 10^{-1}$	$1.74 \times 10^{-2}$	$3.57 \times 10^{-2}$
	Std	$1.80 \times 10^1$	$1.19 \times 10^{-1}$	$4.33 \times 10^{-1}$	$5.14 \times 10^7$	$6.09 \times 10^{-1}$	$1.46 \times 10^{-2}$	$1.91 \times 10^{-2}$
F <sub>13</sub>	Mean	$2.40 \times 10^3$	$9.56 \times 10^{-1}$	$3.44 \times 10^0$	$3.02 \times 10^8$	$1.05 \times 10^0$	$5.03 \times 10^{-2}$	$1.50 \times 10^0$
	Std	$8.78 \times 10^3$	$3.35 \times 10^{-1}$	$9.34 \times 10^{-1}$	$1.46 \times 10^8$	$1.26 \times 10^0$	$3.78 \times 10^{-2}$	$3.27 \times 10^{-1}$
F <sub>14</sub>	Mean	$3.18 \times 10^0$	$3.09 \times 10^0$	$1.23 \times 10^0$	$1.33 \times 10^1$	$1.69 \times 10^0$	$1.43 \times 10^0$	$2.53 \times 10^0$
	Std	$4.17 \times 10^0$	$3.20 \times 10^0$	$1.09 \times 10^0$	$7.01 \times 10^0$	$1.47 \times 10^0$	$9.26 \times 10^{-1}$	$3.36 \times 10^0$
F <sub>15</sub>	Mean	$5.22 \times 10^{-4}$	$1.20 \times 10^{-3}$	$1.51 \times 10^{-3}$	$2.55 \times 10^{-2}$	$1.91 \times 10^{-3}$	$1.06 \times 10^{-3}$	$1.13 \times 10^{-3}$
	Std	$3.15 \times 10^{-4}$	$2.42 \times 10^{-3}$	$1.06 \times 10^{-3}$	$4.00 \times 10^{-2}$	$4.24 \times 10^{-3}$	$2.49 \times 10^{-3}$	$3.66 \times 10^{-3}$
F <sub>16</sub>	Mean	$-1.03 \times 10^0$	$-1.03 \times 10^0$	$-1.03 \times 10^0$	$-9.23 \times 10^{-1}$	$-1.03 \times 10^0$	$-1.03 \times 10^0$	$-1.03 \times 10^0$
	Std	$3.84 \times 10^{-13}$	$7.88 \times 10^{-8}$	$5.76 \times 10^{-16}$	$2.82 \times 10^{-1}$	$4.52 \times 10^{-16}$	$4.68 \times 10^{-16}$	$5.38 \times 10^{-16}$
F <sub>17</sub>	Mean	$1.60 \times 10^{-1}$	$1.62 \times 10^{-1}$	$1.61 \times 10^{-1}$	$1.75 \times 10^{-1}$	$1.61 \times 10^{-1}$	$1.61 \times 10^{-1}$	$1.61 \times 10^{-1}$
	Std	$2.76 \times 10^{-4}$	$3.57 \times 10^{-3}$	$7.99 \times 10^{-4}$	$2.99 \times 10^{-2}$	$3.61 \times 10^{-3}$	$3.60 \times 10^{-3}$	$3.66 \times 10^{-3}$
F <sub>18</sub>	Mean	$3.00 \times 10^0$	$3.00 \times 10^0$	$3.00 \times 10^0$	$3.54 \times 10^1$	$3.90 \times 10^0$	$3.00 \times 10^0$	$3.00 \times 10^0$
	Std	$7.94 \times -12$	$7.37 \times 10^{-4}$	$1.71 \times 10^{-15}$	$1.52 \times 10^2$	$4.93 \times 10^0$	$4.59 \times 10^{-15}$	$3.56 \times 10^{-15}$
F <sub>19</sub>	Mean	$-3.86 \times 10^0$						
	Std	$1.04 \times 10^{-3}$	$9.81 \times 10^{-3}$	$2.49 \times -15$	$7.13 \times 10^{-5}$	$2.13 \times 10^{-15}$	$2.20 \times 10^{-15}$	$2.44 \times 10^{-15}$
F <sub>20</sub>	Mean	$-3.22 \times 10^0$	$-3.27 \times 10^0$	$-3.32 \times 10^0$	$-3.27 \times 10^0$	$-3.31 \times 10^0$	$-3.31 \times 10^0$	$-3.28 \times 10^0$
	Std	$7.30 \times 10^{-2}$	$7.29 \times 10^{-2}$	$2.46 \times 10^{-3}$	$6.22 \times 10^{-2}$	$4.11 \times 10^{-2}$	$3.02 \times 10^{-2}$	$5.70 \times 10^{-2}$
F <sub>21</sub>	Mean	$-6.49 \times 10^0$	$-7.76 \times 10^0$	$-9.27 \times 10^0$	$-5.99 \times 10^0$	$-9.69 \times 10^0$	$-9.65 \times 10^0$	$-9.15 \times 10^0$
	Std	$3.47 \times 10^0$	$2.63 \times 10^0$	$1.84 \times 10^0$	$3.73 \times 10^0$	$1.11 \times 10^0$	$1.04 \times 10^0$	$2.60 \times 10^0$
F <sub>22</sub>	Mean	$-7.38 \times 10^0$	$-7.26 \times 10^0$	$-9.71 \times 10^0$	$-4.14 \times 10^0$	$-9.61 \times 10^0$	$-9.94 \times 10^0$	$-9.67 \times 10^0$
	Std	$3.49 \times 10^0$	$2.96 \times 10^0$	$1.16 \times 10^0$	$2.63 \times 10^0$	$1.67 \times 10^0$	$9.81 \times 10^{-1}$	$2.24 \times 10^0$
F <sub>23</sub>	Mean	$-7.44 \times 10^0$	$-6.37 \times 10^0$	$-1.01 \times 10^1$	$-4.13 \times 10^0$	$-9.91 \times 10^0$	$-1.00 \times 10^1$	$-9.02 \times 10^0$
	Std	$3.67 \times 10^0$	$3.01 \times 10^0$	$5.47 \times 10^{-1}$	$3.03 \times 10^0$	$1.42 \times 10^0$	$1.21 \times 10^0$	$3.10 \times 10^0$

## References

- Deng, Y.; Chen, Y.; Zhang, Y.; Mahadevan, S. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment. *Appl. Soft Comput.* **2012**, *12*, 1231–1237. [\[CrossRef\]](#)
- Ammar, A.; Bennaceur, H.; Châari, I.; Koubaa, A.; Alajlan, M. Relaxed Dijkstra and A\* with linear complexity for robot path planning problems in large-scale grid environments. *Soft Comput.* **2016**, *20*, 4149–4171. [\[CrossRef\]](#)
- Soltani, A.; Tawfik, H.; Goulermas, J.; Fernando, T. Path planning in construction sites: Performance evaluation of the Dijkstra, A\*, and GA search algorithms. *Adv. Eng. Inform.* **2002**, *16*, 291–303. [\[CrossRef\]](#)
- Keskin, M.; Çatay, B. Partial recharge strategies for the electric vehicle routing problem with time windows. *Transp. Res. Part C Emerg. Technol.* **2016**, *65*, 111–127. [\[CrossRef\]](#)
- Keskin, M.; Çatay, B. A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. *Comput. Oper. Res.* **2018**, *100*, 172–188. [\[CrossRef\]](#)

6. Keskin, M.; Laporte, G.; Çatay, B. Electric Vehicle Routing Problem with Time-Dependent Waiting Times at Recharging Stations. *Comput. Oper. Res.* **2019**, *107*, 77–94. [CrossRef]
7. Mousavi, M.; Yap, H.J.; Musa, S.N.; Tahriri, F.; Md Dawal, S.Z. Multi-objective AGV scheduling in an FMS using a hybrid of genetic algorithm and particle swarm optimization. *PLoS ONE* **2017**, *12*, e0169817. [CrossRef]
8. Dorigo, M.; Birattari, M.; Stutzle, T. Ant colony optimization. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–39. [CrossRef]
9. 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]
10. Gen, M.; Lin, L. Multiobjective evolutionary algorithm for manufacturing scheduling problems: State-of-the-art survey. *J. Intell. Manuf.* **2014**, *25*, 849–866. [CrossRef]
11. Shao, X.; Liu, J.; Xu, Q.; Huang, Q.; Xiao, W.; Wang, W.; Xing, C. Application of A Robotic System with Mobile Manipulator and Vision Positioning. In Proceedings of the 2015 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Busan, Republic of Korea, 7–11 July 2015; IEEE: New York, NY, USA, 2015; pp. 506–511. Available online: <https://www.webofscience.com/wos/allldb/full-record/WOS:000381493900087> (accessed on 17 September 2023).
12. Tu, J.C.; Qian, X.M.; Lou, P.H. Application research on AGV case: Automated electricity meter verification shop floor. *Ind. Robot. Int. J. Robot. Res. Appl.* **2017**, *44*, 491–500. [CrossRef]
13. Hu, E.; He, J.; Shen, S. A dynamic integrated scheduling method based on hierarchical planning for heterogeneous AGV fleets in warehouses. *Front. Neurobotics* **2023**, *16*, 1053067. [CrossRef]
14. Xin, B.; Lu, S.; Wang, Q.; Deng, F.; Shi, X.; Cheng, J.; Kang, Y. Simultaneous Scheduling of Processing Machines and Automated Guided Vehicles via a Multi-View Modeling-Based Hybrid Algorithm. *IEEE Trans. Autom. Sci. Eng.* **2023**, 1–15. [CrossRef]
15. Niu, H.; Wu, W.; Xing, Z.; Wang, X.; Zhang, T. A novel multi-tasks chain scheduling algorithm based on capacity prediction to solve AGV dispatching problem in an intelligent manufacturing system. *J. Manuf. Syst.* **2023**, *68*, 130–144. [CrossRef]
16. Martin, X.A.; Hatami, S.; Calvet, L.; Peyman, M.; Juan, A.A. Dynamic Reactive Assignment of Tasks in Real-Time Automated Guided Vehicle Environments with Potential Interruptions. *Appl. Sci.* **2023**, *13*, 3708. [CrossRef]
17. Wang, Z.; Wu, Y. An Ant Colony Optimization-Simulated Annealing Algorithm for Solving a Multiload AGVs Workshop Scheduling Problem with Limited Buffer Capacity. *Processes* **2023**, *11*, 861. [CrossRef]
18. Meng, L.; Cheng, W.; Zhang, B.; Zou, W.; Fang, W.; Duan, P. An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem. *Sensors* **2023**, *23*, 3815. [CrossRef]
19. Gao, Y.; Chen, C.-H.; Chang, D. A Machine Learning-Based Approach for Multi-AGV Dispatching at Automated Container Terminals. *J. Mar. Sci. Eng.* **2023**, *11*, 1407. [CrossRef]
20. Liu, Q.; Wang, C.; Li, X.; Gao, L. An improved genetic algorithm with modified critical path-based searching for integrated process planning and scheduling problem considering automated guided vehicle transportation task. *J. Manuf. Syst.* **2023**, *70*, 127–136. [CrossRef]
21. Wang, Z.; Zeng, Q. A branch-and-bound approach for AGV dispatching and routing problems in automated container terminals. *Comput. Ind. Eng.* **2022**, *166*, 107968. [CrossRef]
22. Adamo, T.; Ghiani, G.; Guerriero, E. Recovering feasibility in real-time conflict-free vehicle routing. *Comput. Ind. Eng.* **2023**, *183*, 109437. [CrossRef]
23. Zhou, B.-H.; Zhang, J.-H. An improved bi-objective salp swarm algorithm based on decomposition for green scheduling in flexible manufacturing cellular environments with multiple automated guided vehicles. *Soft Comput.* **2023**, *27*, 16717–16740. [CrossRef]
24. Zou, W.-Q.; Pan, Q.-K.; Meng, T.; Gao, L.; Wang, Y.-L. An effective discrete artificial bee colony algorithm for multi-AGVs dispatching problem in a matrix manufacturing workshop. *Expert Syst. Appl.* **2020**, *161*, 113675. [CrossRef]
25. Jiang, Z.; Zhang, X.; Wang, P. Grid-Map-Based Path Planning and Task Assignment for Multi-Type AGVs in a Distribution Warehouse. *Mathematics* **2023**, *11*, 2802. [CrossRef]
26. Wu, N.; Zhou, M. Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **2005**, *35*, 1193–1202. [CrossRef]
27. Singh, N.; Dang, Q.-V.; Akcay, A.; Adan, I.; Martagan, T. A matheuristic for AGV scheduling with battery constraints. *Eur. J. Oper. Res.* **2022**, *298*, 855–873. [CrossRef]
28. Boccia, M.; Masone, A.; Sterle, C.; Murino, T. The parallel AGV scheduling problem with battery constraints: A new formulation and a matheuristic approach. *Eur. J. Oper. Res.* **2023**, *307*, 590–603. [CrossRef]
29. Abderrahim, M.; Bekrar, A.; Trentesaux, D.; Aissani, N.; Bouamrane, K. Manufacturing 4.0 Operations Scheduling with AGV Battery Management Constraints. *Energies* **2020**, *13*, 4948. [CrossRef]
30. Dang, Q.-V.; Singh, N.; Adan, I.; Martagan, T.; van de Sande, D. Scheduling heterogeneous multi-load AGVs with battery constraints. *Comput. Oper. Res.* **2021**, *136*, 105517. [CrossRef]
31. Li, J.; Tang, W.; Zhang, D.; Fan, D.; Jiang, J.; Lu, Y. Map Construction and Path Planning Method for Mobile Robots Based on Collision Probability Model. *Symmetry* **2023**, *15*, 1891. [CrossRef]
32. Hashim, F.A.; Hussien, A.G. Snake Optimizer: A novel meta-heuristic optimization algorithm. *Knowl.-Based Syst.* **2022**, *242*, 108320. [CrossRef]
33. Long, W.; Jiao, J.; Liang, X.; Cai, S.; Xu, M. A Random Opposition-Based Learning Grey Wolf Optimizer. *IEEE Access* **2019**, *7*, 113810–113825. [CrossRef]

34. Tang, G.; Tang, C.; Claramunt, C.; Hu, X.; Zhou, P. Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment. *IEEE Access* **2021**, *9*, 59196–59210. [[CrossRef](#)]
35. Yao, X.; Liu, Y.; Lin, G. Evolutionary programming made faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–102. [[CrossRef](#)]

**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.