Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 32(3): 1946–1972.
DOI: 10.3934/era.2024089
Received: 22 January 2024
Revised: 19 February 2024
Accepted: 22 February 2024
Published: 06 March 2024

Research article

Design of a novel multimodal optimization algorithm and its application in logistics optimization

Weishang Gao[1], Qin Gao[1,*], Lijie Sun[2] and Yue Chen[1]

[1] School of Intelligent Manufacturing, Taizhou University, Taizhou 318000, China
[2] School of Electronics and Information Engineering, Taizhou University, Taizhou 318000, China

* **Correspondence:** Email: wei_qin@foxmail.com.

Abstract: This study was purposed to design a multimodal continuous optimization algorithm based on a scheme agent to address the multidimensional complexity of optimization. An evolutionary sampling method of subarea exploration and multiple exploitations was developed by employing the scheme with variable population size so as to obtain higher optimization speed and accuracy. Second, the distribution plan was quantified into high-dimensional variable parameters based on the characteristics of logistics distribution optimization problems, and a high-dimensional discrete optimization model was constructed. Then, we identified and addressed the prominent issues and malignant virtual changes in the application of continuous algorithms to discrete problems. We have introduced a reasonable mutation mechanism during the optimization sampling process to mitigate this issue. Continuous real coordinate points were transformed across the neighborhood to standard discrete integer coordinate points by normalizing and logicizing the optimization sampling coordinates; also, the discretization of the continuous algorithm was realized. This approach could effectively prevent the algorithm from searching for targets in continuous optimization space, thereby fully reducing the complexity of the objective function distribution after conversion. The experiments showed that the transformed multimodal discrete optimization algorithm effectively addressed the optimization design problem of logistics distribution.

Keywords: discrete optimization; logistics distribution; optimization algorithm

1. Introduction

In the current context of rapid economic development and escalating energy scarcity, the improvement of modern logistics systems [1,2], with a particular focus on enhancing green logistics capabilities [3,4], holds significant implications for the promotion of economic growth. In the logistics process, considerations for energy conservation and environmental protection typically involve customizing distribution schemes that feature the shortest path and minimal resource utilization, thereby minimizing transportation costs. This issue was addressed by establishing corresponding combination optimization models [5,6] and developing optimization algorithms [7,8].

In addressing logistics distribution challenges [9–11], it is essential to comprehensively consider various factors such as cargo loading layout [12], vehicle service arrangement [13], transportation path selection [14], loading capacity limitations [15], customer time requirements [16], maintenance costs [17], and operation modes [18,19]. For the sake of research convenience, specific links are generally designed under certain assumptions. For instance, Wang et al. developed a dual-objective optimization model with the aim to maximize the use of distribution vehicle space and minimize logistics operating costs for the loading and distribution of goods. They proposed a hybrid intelligent optimization algorithm to effectively determine the model solution [20]. Huang constructed a multiobjective optimization model for cross-border e-commerce, considering both the timeliness and cost of logistics delivery, and she employed the chicken swarm algorithm for solution analysis [21]. Wu and Yang established a high-dimensional optimization model for vehicle scheduling in logistics and introduced an improved particle swarm optimization (PSO) algorithm to enhance the speed and accuracy of solving the optimization problem [22]. Lu addressed the scheduling problem of Internet-of-Things delivery vehicles and employed a PSO algorithm with nonlinear perturbation factors to enhance the optimization speed and success rate and reduce default penalty costs [23]. Evidently, the functional model created for logistics distribution problems exhibits high complexity, and swarm intelligence optimization algorithms [24] offer a relatively straightforward and effective approach to tackling such intricate optimization problems.

Inspired by bionics [25] and simulations [26], the swarm intelligence optimization algorithms leverage individual inspiration, group sampling, and evolutionary iteration to generate a unique form of intelligence within a collective swarm [27]. This collective intelligence is particularly adept at solving complex optimization problems. Notable examples include the genetic algorithm (GA), which mimics biological heredity and mutation by discretizing decision variables into “chromosomes”, iteratively evolving better decision variables [28,29]. Similarly, PSO-based coordinate decision variables iteratively refine coordinate variables by simulating the foraging behavior of bird flocks, gradually converging toward the optimal position [30,31]. These algorithms possess diverse mechanisms and characteristics, each tailored to address various types of optimization problems [32].

This study delves into the complex challenges associated with optimizing logistics distribution, particularly in terms of dealing with its multidimensional complexity. First, a robust multimodal continuous optimization (MCO) algorithm was developed, emphasizing its ability to effectively explore complex high-dimensional optimization functions. Second, a model suitable for swarm intelligence optimization algorithms was created. This involved breaking down the distribution scheme into high-dimensional decision parameters and establishing the relationship between the distribution costs and these parameters within the objective function. Then, a practical mutation mechanism was introduced, recognizing the potential issue of malignant virtual changes (MVCs) stemming from continuous decision variables. This mechanism converted continuous decision variables into standard discrete ones, addressing the challenge and simplifying the distribution of the objective function. The superior capabilities of the transformed multimodal discrete optimization (MDO) algorithm were confirmed through extensive testing of nominal and case functions, successfully minimizing the delivery cost globally.

2. Related works

The multimodal optimization (MO) algorithm developed in this study is a novel swarm intelligence optimization algorithm that includes both MCO and MDO. Although the proposed MO algorithm also employs heuristic swarm optimization methods, it is different from the heuristic mode of previous similar algorithms. First, our proposed MO algorithm uses a three-layer agent pattern to enable heuristic population sampling in a hierarchical and partitioned manner. Second, the inspired population sampling pattern involves density evolution that is guided by range and quantity. Finally, a rational mutation (RM) mechanism has been introduced into the discrete optimization to eliminate MVCs in the objective function.

2.1. Overconvergence

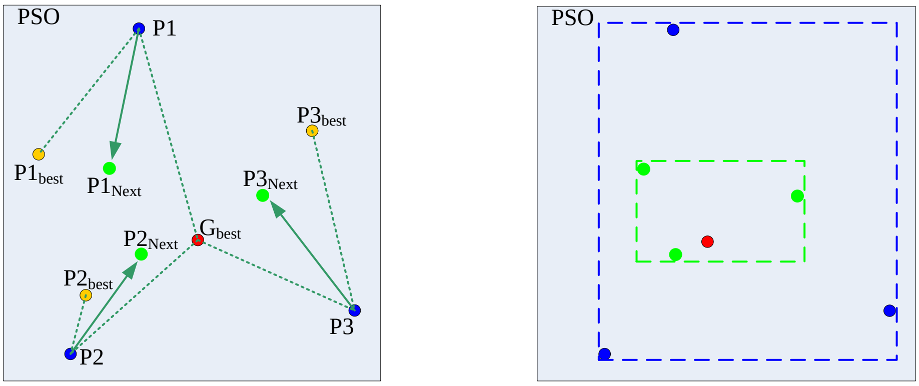
PSO is a swarm intelligence optimization algorithm with fast convergence and more flexible changes [30,31]. Its core dynamic update mechanism is shown in Eq (2.1).

$$\begin{cases} \vec{v}_{i+1} = \omega\vec{v}_i + \vec{U}\left(0,\phi_1\right)\otimes\left(\vec{p}_i - \vec{x}_i\right) + \vec{U}\left(0,\phi_2\right)\otimes\left(\vec{p}_g - \vec{x}_i\right) \\ \vec{x}_{i+1} = \vec{x}_i + \vec{v}_{i+1} \end{cases} \tag{2.1}$$

where $\vec{v}_{i+1}$ and $\vec{v}_i$ represent the steps of the current individual in the (i+1)-th and i-th generations, respectively; $\vec{x}_{i+1}$ and $\vec{x}_i$ represent the sampling positions of the current individual in the (i+1)-th and i-th generations, respectively; ω represents the system inertia, particularly the degree to which the current individual maintains the original step size; the function $\vec{U}\left(0,\phi\right)$ generates a random weight coefficient from 0 to ϕ, increasing the diversity of the swarm; $\otimes$ represents the influence of the weight coefficient and is generally treated as multiplication; and $\vec{p}_i$ $\vec{p}_g$ represent the personal best positions of the current individual and the global best position of the current swarm, respectively. Based on the aforementioned dynamic evolution formula, individual sampling directions of particles can be updated one by one based on both personal best and global best information.

On the one hand, heuristic information originates from a global optimum and an individual optimum, effectively improving population diversity. However, as the heuristic processing is applied globally, it tends to push the population toward local optima. This is the fundamental reason why PSO tends to fall into local optima. On the other hand, the evolutionary process of the particle swarm is formed by moving individual particles, and the movement here is not restricted by the partition. When the population is inspired by optimal information, the algorithm achieves large-scale and fast convergence toward the optimal region. However, this is also why the algorithm may fall into a local optimum.

We constructed a dynamic analysis diagram for the particle swarm algorithm (Figure 1) to enable a more intuitive understanding of the convergence problem. The distribution and updating of particle points were based on the dynamic updating method of the PSO algorithm. The blue dots in the figure represent individuals P1, P2, and P3 of the current particle, and their corresponding personal best points $P1_{best}$, $P2_{best}$, and $P3_{best}$, respectively, are represented by yellow dots. The red dots represent the current global best point Gbest. Each current individual is inspired and guided by their personal best point and the current global best point to generate the positions of the next generation of particles, such as $P1_{Next}$, $P2_{Next}$, and $P3_{Next}$, represented by the green dots in the figure.



(a)

(b)

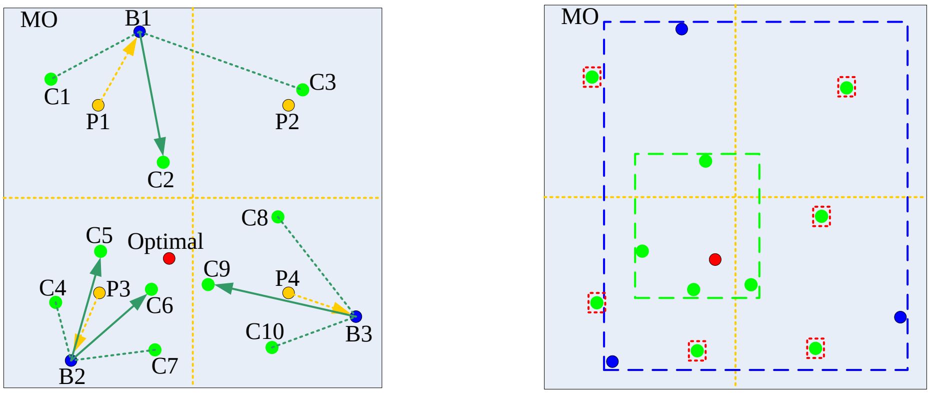
Figure 1. Dynamic distribution of sampling in the PSO algorithm.
((a) shows how the particle swarm is updated. (b) shows the convergence of the swarm boundary.)

On the one hand, each particle is attracted to G_{best}, which directly influences the convergence of all particles toward a single point. On the other hand, each particle is also attracted to its personal best point, which has a certain proximity relationship with G_{best} and indirectly contributes to the convergence of all particles toward a single point. Based on the aforementioned two aspects of analysis, the entire particle swarm can quickly converge to a certain optimal region. Figure 1(b) shows the boundary lines of the two generations of the particle swarm, corresponding to the blue dashed box and green dashed box, respectively. The swarm boundary shows a clear converging trend, consistent with the aforementioned results of analysis.

Although the dynamic distribution of the optimized population sampling has the characteristic of efficient convergence, this mechanism increases the possibility of falling into local optima due to overconvergence. After an in-depth analysis, we further found that the mechanism by which the entire population can converge toward a single point is a key factor in the formation of the above-mentioned characteristics.

In view of this, we used a multimodal approach to introduce hierarchical and partitioned processing modes to the population, which reduced the possibility of falling into local optima. In addition, a partition restriction on the population evolution was added to the MCO and MDO algorithms to ensure global exploration capabilities.

We constructed a dynamic schematic diagram of the corresponding sampling optimization method (Figure 2) to allow us visually understand the inhibitory effect of grading and partitioning on the overconvergence phenomenon. The yellow dots denoted by P1, P2, P3, and P4 in the figure represent four partitions, divided by yellow dashed lines. Based on the partition agents, basic individuals B1, B2, and B3 were generated within the corresponding partitions. Inspired by the basic individuals and partition agents, we generated sub-generation individuals C1–C10 in the corresponding and surrounding partitions. The optimal sub-generation individuals were selected to transform into basic individuals in the next iteration. The P2 partition in the figure does not continue to output during grading, indicating that the fitness of this partition was too low and the speed of new individual generation was limited to a certain extent. Basic individuals were limited by partition agents, making the sampling optimization points as dispersed as possible. The sub-generation individuals were influenced by the basic individuals, limiting the tendency of the group to converge toward a single point. Therefore, the dynamic distribution of the optimized population in the new MO algorithm was determined by the hierarchical and partitioned patterns. This approach aimed to overcome the problem of overconvergence of the global population to a single point, while still retaining the advantage of the particle swarm quickly converging to a more optimal region.



(a) (b)

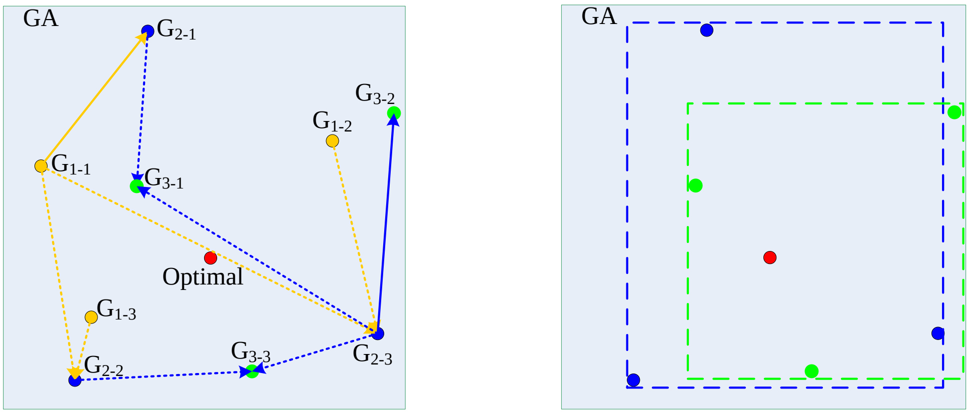
Figure 2. Dynamic distribution of sampling in the MO algorithm.
((a) shows how the individuals are updated. (b) shows the distribution and boundaries of samples.)

The distribution and boundaries of the two-level populations are shown in Figure 2(b). The number of individuals in the two levels was different, and the number of basic individuals in the two iterations was different after the better sub-generation individuals were converted to the basic individuals in the next iteration. This application of a variable population size provided a more flexible convergence method for the new algorithm. The basic boundary and the sub-generation optimal boundary are represented by blue dashed boxes and green dashed boxes, respectively. Although the boundary had a converging trend that is similar to that of the PSO algorithm, the boundary convergence in the MO algorithm is limited by partitioning. The contraction of the sub-generation optimal boundary is based on the survival of the fittest among the sub-generation individuals. The eliminated individuals are shown as dots with red dashed boxes in the figure, and they play a role in global exploration and indirectly limit the overconvergence of the swarm. Figure 2 only illustrates the dynamic updating phenomenon of the MO algorithm, and its detailed dynamic mechanism is introduced in Chapter 3.

2.2. Efficiency

The GA is a swarm intelligence optimization algorithm with a wide search range and expansive population changes. Its core dynamic updating mechanism is based in the crossover and mutation of chromosomes [28,29]. On the one hand, the crossover of excellent individuals preserves some superior features while avoiding excessive convergence in the dominant region. On the other hand, individual random mutation preserves the global exploration mechanism throughout the evolution process. Although these factors improve the optimization reliability of the GA, the evolutionary direction of the individuals after their crossover and mutation is not easy to control, which introduces some difficulties that can affect the algorithm's convergence.

We constructed a dynamic updating diagram for individuals (Figure 3) as per the GA and based on the studies of crossover and mutation mechanisms. The yellow dots represent the previous-generation individuals G_{1-1}, G_{1-2}, and G_{1-3}. G_{1-1} and G_{1-2} cross to generate the current-generation individual G_{2-3}. G_{1-1} and G_{1-3} cross to generate the current-generation individual G_{2-2}. G_{1-1} mutates to generate the current-generation individual G_{2-1}. The relationships of generation are shown by yellow dashed arrows in the figure. Similarly, the current generation of individuals generates the next generation of individuals G_{3-1}, G_{3-2}, and G_{3-3} through crossover and mutation, and the corresponding relationships of generation are shown by blue dashed arrows in the figure.



(a) (b)

Figure 3. Dynamic distribution of sampling in the GA.
((a) shows how the individuals are updated. (b) shows the changing of distribution boundaries.)

The heuristic effect of the global optimal position, represented by red dots in Figure 3, on the population is not particularly significant. The main reason for this is the relatively strong diversity of new populations that are generated through crossover and mutation. As shown in Figure 3(b), the boundaries of the two generations are represented by blue and green dashed boxes, respectively. The convergence speed between the two generations was not as fast as that for the previous PSO and MO algorithms, but the dispersion of individual distribution increased the reliability of the algorithm. On the other hand, new individuals generated through a single crossover or mutation may not necessarily be better, but they appeared to be more comprehensive in the global search. Based on the aforementioned analysis, the distribution of the population in the GA was found to be relatively scattered, and the overall density was relatively uniform. This provided an effective basis for us to improve the efficiency of the new algorithm by using the density of the optimized population.

In this study, density control was applied in the evolutionary process of MO, and the distribution of individuals was restricted by partition range. This not only effectively circumvented the excessive loss of excellent individuals, it also facilitated the control of the regional distribution of population convergence.

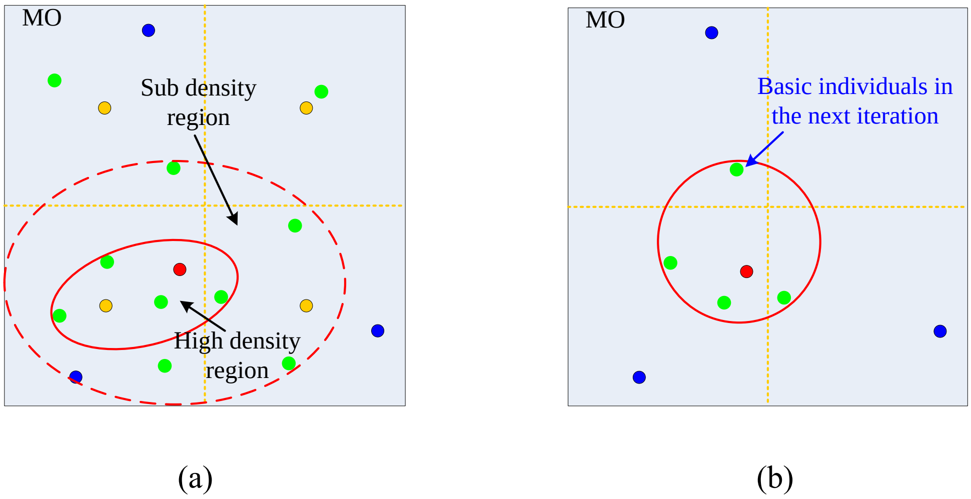


Figure 4. Density distribution of sampling in the MO algorithm.
((a) shows how the individuals gather. (b) shows the iterative convergence.)

We found that introducing hierarchical and partitioned mechanisms into the sampling optimization process not only solved the problem of overconvergence, it also enabled flexible adjustment of the sampling density according to the different distributions of the objective function. Figure 4 presents diagrams that illustrate the results of analyzing the distribution of sampling points in the MO algorithm. The information at each point in the figure was found to be consistent with that in Figure 2, where the eliminated sub-generation individuals in the right graph are no longer displayed. The area enclosed within the solid red circle in Figure 4(a) is the high-density sampling area, and the area enclosed within the dashed red circle is the less high density sampling area. In Figure 4(b), the green dots that were converted to basic individuals in the next iteration converged to the optimal target area over iterations, further increasing the sampling density nearby the optima. After analysis, the higher sampling density in the optimal target area was found to be due to two reasons. First, after hierarchical and partition processing, more individuals were generated as sub-generations in the more effectively optimized partition, forming a partition with a higher sampling density. Second, after hierarchical and partition processing, the distribution of individuals in a new sub-generation layer were found to spread toward the more optimal neighboring area. Because of the aforementioned two reasons, a trend of sampling convergence could be observed near the optimal target, providing a basis for improving the effectiveness of swarm optimization algorithms.

2.3. Discretization

Mapping continuous independent variables into discrete variables by changing the computation of the objective function is a common technique that is used to transform a discrete optimization problem into a continuous optimization problem [33]. Although this method provides the necessary conditions for the application of continuous optimization algorithms to discrete optimization problems, it can sometimes result in a more complex distribution of the modified objective function.

This phenomenon is discussed in detail in Section 3.2, and the concept of MVCs is also explained. To address this issue, we suggest the use of an RM mechanism based on the characteristics of MO that discretizes the decision variables represented by sampled individuals. The results of analysis show that this approach could eliminate the MVC phenomenon.

3. Algorithm design

3.1. Framework design of MCO

The heuristic optimization algorithm involves global exploration, local mining, and the transition between these phases. These mechanisms are evident in the distribution characteristics of agents in the optimization space, corresponding to global dispersion, local aggregation, and the transition between dispersion and aggregation, respectively. Consequently, as the optimization process iterates, the results accumulated as a result of successive generations of population sampling were found to exhibit varying density distributions, as depicted in the process shown in Figure 5. At a local level, the higher the fitness of a region, the greater the distribution density of the optimization agents, and vice versa. On a global scale, this process of density distribution change is determined by the population evolution pattern and the utilization of individual sampling information, constituting a key factor that influences the accuracy and speed of the optimization algorithm.

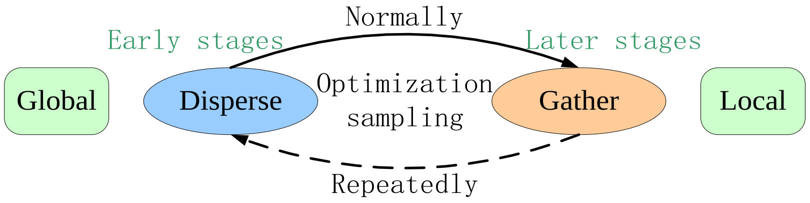


Figure 5. Mode conversion for swarm optimization sampling.

The MO algorithm proposed in this paper primarily uses local uniform sampling with varying densities in different ranges. It gradually changes globally optimized sampling points from exhibiting dispersion to aggregation over successive generations. This approach fundamentally differs from conventional swarm intelligence optimization algorithms, as depicted in Figure 6. The key to influencing the accuracy and efficiency of the optimization process lies in whether this transition aligns with the problem of fitness distribution. For instance, in scenarios with a monotonic target fitness variation, minimizing the waste of sampling resources due to scattered sampling in disadvantaged areas enhances the progression from scattered to aggregated, consequently expediting the emergence of optimal results. Conversely, in cases with complex target fitness variation, the transition from scattered to aggregated progresses at a slower rate. In such scenarios, a broader uniform multiple sampling range was employed to precisely capture the optimal region, thereby enhancing the accuracy of the optimal result. Effectively evaluating the distribution characteristics of optimization space was therefore pivotal in ensuring the quality of information judgment.

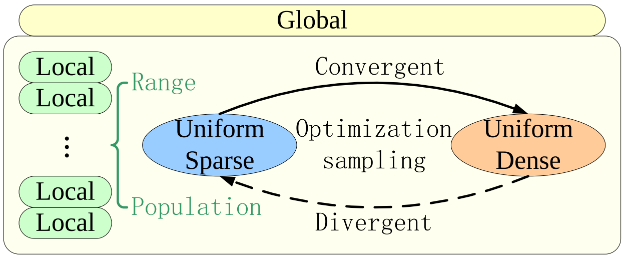


Figure 6. Dynamic sampling mechanism of the MO algorithm.

In fact, the distribution status and characteristics of change of the target fitness in the global region are not limited to the two aforementioned cases, especially in different regions. The complex and diverse changes in fitness require more flexible algorithm to analyze and judge during the sampling process. Therefore, we propose to divide the heuristic optimization mechanism into three levels, as shown in Figure 7. Different sampling behaviors were applied to realize the transitions between different levels.

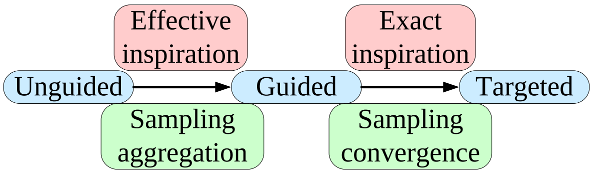


Figure 7. Three-layer heuristic optimization behavior.

We introduced a multiangle sampling information processing method to better understand the current sampling information and improve its heuristic effectiveness, along with the corresponding adjustment strategy outlined in Table 1. Automatically adjusting the swarm sampling density based on the sampling range and sample count allows the algorithm to strike a balanced trade-off between exploration and exploitation. This globally integrated regulation of the sampling density provides a more reliable mechanism for intelligent algorithms to enhance their adaptive evolutionary properties. Notably, as the optimization iterations progressed, even in regions with temporarily low levels of target fitness, sampling persisted, albeit with a relatively smaller sample count. Furthermore, with an increase in the fitness of a region, the number of sampling points was found to rapidly increase, demonstrating significant potential to prevent the optimized population from falling into a local optimum.

Table 1. Comparison of sampling information and strategy adjustment results.

	Fitness		Fitness growth	
	High	Low	Fast	Slow
Sampling range	Decrease	Increase	-	-
Samples number	-	-	Increase	Decrease

3.2. MDO discretization

It is essential to map continuous decision variables to discrete ones during the fitness calculation process to adapt the continuous optimization algorithm for discrete optimization problems. This process involves not only creating the decision variables to suit specific problems, but also facilitating the process of variable updating that could originally take continuous values within the discrete decision space of the objective function. This transformation effectively converts the discrete objective function into a continuous one.

3.2.1. Decision variables

Regarding the optimization of multisite and multivehicle cargo distribution, different allocations of transportation vehicles and delivery orders for destination sites can yield varied tariff results. We introduced the concept of distribution genes as decision variables (agents) to enhance the clarity of the distribution scheme representation. This approach aligns with the information recording feature of GA chromosomes. As shown in Figure 8, N_v and N_s represent the quantity of vehicles and sites, respectively, and V_i and S_i represent the number of vehicles and sites corresponding to the i-th node, respectively. The gene sequence was set as twice the length of the number of sites (N_s sites) and divided into two parts. The first part was assigned a vehicle value (N_v vehicles) for each site within the range of (0, N_v], totaling N_s gene elements, each corresponding to a site. The latter part was assigned a service order value for each site within the range of (0, N_v), totaling N_s gene elements, also corresponding to each site.

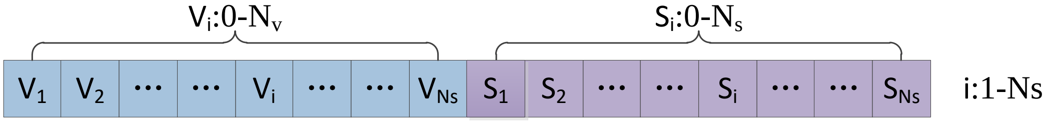


Figure 8. Distribution of gene sequences.

The unique assignment of the vehicle value gene element to each site ensured that only one vehicle was designated for delivering goods to each site. This approach allows each vehicle to be assigned to and serve different sites, preventing duplicate allocation of vehicle resources. During the goods distribution process, each vehicle distributes goods in ascending order based on the site numbers among the sites that it is tasked to serve at that time.

The distribution of gene sequences depicted in Figure 8 functioned as a decision vector in a multidimensional optimization problem. In this vector, each element represents a decision variable in one dimension. The values of these variables within the distributed gene sequences, i.e., distribution genes, changed continuously, allowing agents to adjust their spatial coordinates based on heuristic factors. This mechanism sustained population diversity in continuous optimization problems, enhancing the overall optimization search accuracy.

3.2.2. Fitness calculation

First, continuous decision variables were converted into discrete decision variables. Vehicle values were rounded up to obtain the corresponding vehicle numbers, that is, integers within the interval [1, N_v]. Stations with the same vehicle number formed a group, and the algorithm sorted all stations in the group based on their order values (real numbers). The smaller the order value of a site, the higher its distribution order within the group. As shown in Figure 9, the first four and the last four sites were assigned to car 1 and car 2, respectively. Within each group, the first site held the highest distribution priority, followed by the next three in descending order. The corresponding stations in the figure were sequentially assigned an order number (integer) within the group to simplify the representation of the distribution order.

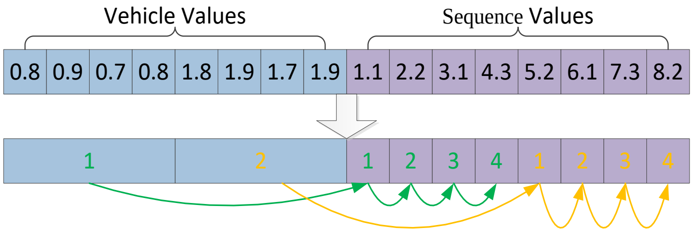


Figure 9. Discretization of continuous distribution genes.

Second, continuous decision variables were converted into discrete decision variables. The vehicle values were rounded up to obtain corresponding vehicle numbers, that is, integers within the interval [1, Nv]. Stations sharing the same vehicle number formed a group, and the algorithm sorted all stations in a group based on their order values (real numbers). Within each group, the distribution order was determined by the order values, with lower values indicating higher distribution priority. As illustrated in Figure 9, car 1 served the first four sites, while car 2 served the last four sites. Within each group, the first site held the highest distribution priority, followed by the next three in descending order. For clarity in distribution order representation, each station in the figure was sequentially assigned an order number (integer) within its group.

Finally, for the distribution scheme that met the constraint requirements, intra-group distribution costs were calculated based on the distribution order of each group. The sum of the costs of each group was then used as the objective function value for the current decision variables. We applied a table lookup strategy based on the table of intersite distribution costs to enhance the algorithm speed. A sample table of 10 intersite distribution costs is shown in Table 2, where the freight center is listed as site 0 and other sites are named in order starting from 1, respectively. The table was reflected in the program as a cost matrix **F**, and its internal elements **F**(m, n) indicated the cost of transferring the goods from site $m - 1$ to site $n - 1$. For example, the in-group distribution cost of vehicle 1 shown in Figure 9 was **F**(1, 2) + **F**(2, 3) + **F**(3, 4) + **F**(4, 5).

3.2.3. Problem analysis

The aforementioned scheme was designed to compute discrete objective functions by mapping continuous decision variables to discrete ones within the objective function. This maintained the continuity of the independent variables outside of the objective function, allowing for direct application of the continuous optimization algorithm to the discrete optimization problem. However, this scheme might complicate the distribution of the objective function for the sorting section of the distribution optimization problem.

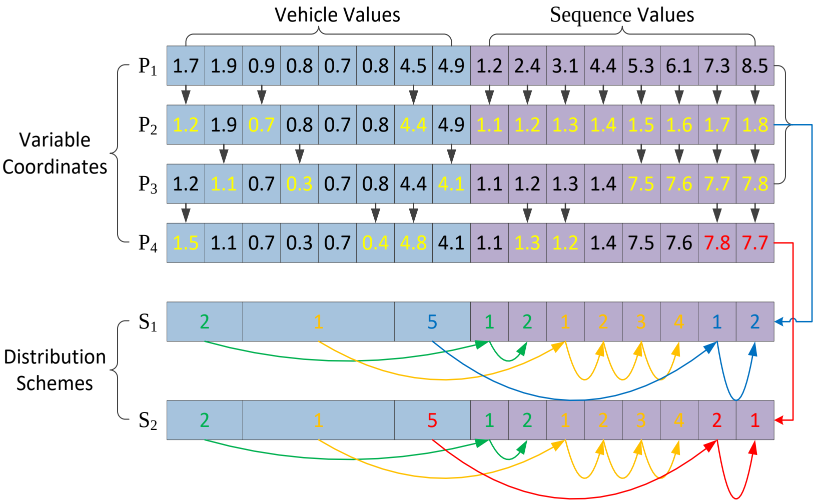

Vehicle Values
Sequence Values
Variable Coordinates
P1 1.7 1.9 0.9 0.8 0.7 0.8 4.5 4.9 1.2 2.4 3.1 4.4 5.3 6.1 7.3 8.5
P2 1.2 1.9 0.7 0.8 0.7 0.8 4.4 4.9 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8
P3 1.2 1.1 0.7 0.3 0.7 0.8 4.4 4.1 1.1 1.2 1.3 1.4 7.5 7.6 7.7 7.8
P4 1.5 1.1 0.7 0.3 0.7 0.4 4.8 4.1 1.1 1.3 1.2 1.4 7.5 7.6 7.8 7.7
Distribution Schemes
S1 2 1 5 1 2 1 2 3 4 1 2
S2 2 1 5 1 2 1 2 3 4 2 1

Figure 10. Example of objective function complexity.

As depicted in Figure 10, four continuous decision variables, named P_1, P_2, P_3, and P_4, exist. These decision variables could be transformed into two discrete strategies, named S_1 and S_2. For the three continuous decision variables P_1, P_2, and P_3, their agent coordinates were found to be far apart in the optimization space. However, the discrete distribution solutions they represented were the same; that is, the objective function values were of the same magnitude. Conversely, the corresponding coordinates between P_3 and P_4 were found to be close to each other, but the discrete distribution schemes they represented were not the same. Analogously, a small change in the order of the distribution genes as an independent variable might cause a significant change in the distribution cost, making the distribution of the continuous objective function exceptionally complex due to the influence of the sorting process. This small change in the independent variable should not exist in the discrete problem. Therefore, we chose to refer to the complexity of the objective function caused by the small change in the delivery gene variable as an MVC.

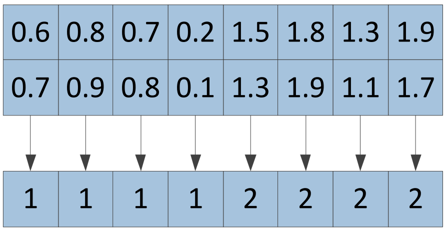
3.3. RM

We have introduced an RM mechanism that allows agent coordinates to circumvent the complexities of continuous objective functions. This mechanism restricts independent variables within the discrete decision space, effectively eliminating MVCs that may result from continuous minor alterations in the distribution genes. Specifically, in optimization problems involving sorting, consecutive value sorting can lead to the manifestation of MVCs. Thus, we categorized the introduced

RM mechanism for the logistics distribution problem into two components: value mutation and sequencing mutation. These components were designed to manage the discretization of vehicle values and sequential values, respectively.

3.3.1. Value mutation

Value mutation commonly involves rounding decimals to the nearest integers. For ease of mechanism analysis and program readability, we employed upward rounding to convert vehicle values from the continuous real-number domain to discrete integer vehicle numbers. Assuming that N_v vehicles were available for assignment, within the continuous optimization algorithm, vehicle values could freely range in the interval $(0, N_v]$, and the result obtained after upward rounding yielded vehicle numbers in the set $\{1, 2, ..., N_v\}$. Because the length of the continuous real-number space corresponding to each vehicle number was equal, the probability of individual vehicle numbers being assigned was equal when values were randomly selected within the set of real numbers.



0.6	0.8	0.7	0.2	1.5	1.8	1.3	1.9
0.7	0.9	0.8	0.1	1.3	1.9	1.1	1.7
1	1	1	1	2	2	2	2

Figure 11. Example of value mutation.

Figure 11 depicts the assignment of two vehicles to eight stations. The two assignment outcomes within the set of real numbers are presented in the first two rows of Figure 11 and were found to fall within the range of real numbers (0, 2]. Following the value mutation, the vehicle number assigned to the first four stations becomes 1, while the vehicle number assigned to the last four stations becomes 2. This sample demonstrated that the results of the assignment in the real-number domain differed, but after value mutation, they could be converted to the same discrete results in the nearby integer domain. This transformation essentially converted multi-point sampling within a local region in the continuous optimization process into a single repeated sampling process at a point in the discrete optimization process. This complicated the process of transitioning continuous optimization algorithms to discrete optimization processes. One viable approach to mitigate this issue was to limit the sampling density. This helped to prevent the excessive waste of computational resources due to repeated sampling.

3.3.2. Sequencing mutation

The influence of site grouping was excluded from the introduced sequencing mutation to maintain clarity in the distribution genes and mitigate the impact of site grouping. First, sequence values for all sites were directly prioritized. Second, corresponding sites were assigned integers from 1 to N_s to denote their preliminary sequence number, that is, the priority number, based on the priority ranking from smallest to largest. Finally, the sites were grouped according to the assigned vehicles, and the delivery order within each group followed the priority number, ranging from smallest to largest.

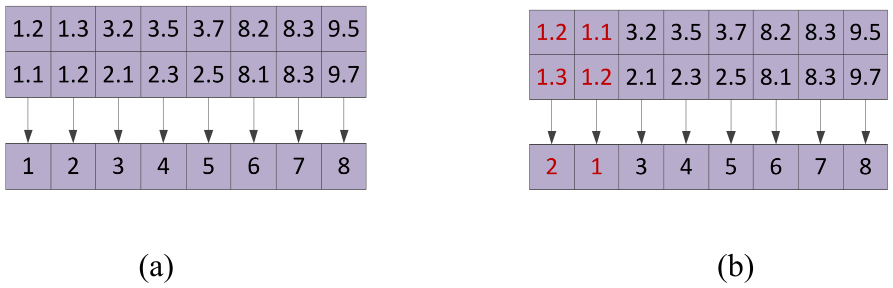


Figure 12. Example of sequencing mutation.
((a) Original distribution genes. (b) Mutant distribution genes.)

Figure 12 illustrates different scenarios for prioritizing eight sites. The four sorting outcomes within the set of real numbers are presented in the first two rows of Figure 12(a),(b), showcasing flexible and variable values. After sequencing mutation, it was observed that the left two sorting methods converged into one result, and the right two sorting methods converged into another result, resembling the characteristics of the value mutation. Additionally, comparing the sorted values of Figure 12(a),(b) real-number sets revealed that the differences between them were not significant, and the distance reflected in the coordinate space was also small. However, the order of relationships they represented differed. Through sequencing mutation, this sorting discrepancy was effectively visualized and the position coordinates were separated in the discrete optimization space. Clearly, after the sequencing mutation process, the slight changes in the distribution gene variables that could induce sorting changes in continuous space were reflected in the discrete space and the changes in the discrete distribution gene variables were evident, building a foundation to avoid the occurrence of MVC.

After the two RM processes of value mutation and sequencing mutation, the distribution genes of P_1, P_2, and P_3 in Figure 10 were converted into integer variables. As illustrated in Figure 13, the coordinates of their positions in the decision space were shifted to the integer coordinate point P_5, transforming decision variables from the continuous real set to the discrete integer set in the optimization algorithm.

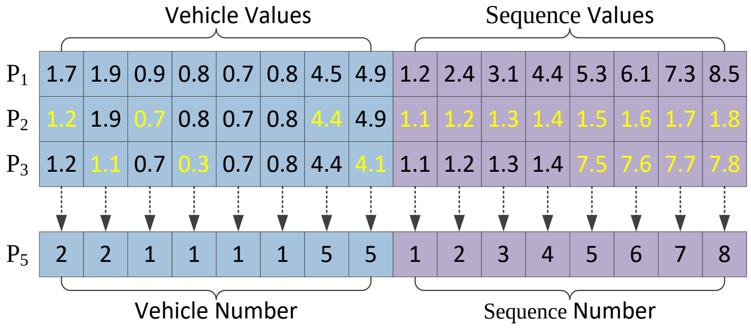

Vehicle Values
Sequence Values
P1 1.7 1.9 0.9 0.8 0.7 0.8 4.5 4.9 1.2 2.4 3.1 4.4 5.3 6.1 7.3 8.5
P2 1.2 1.9 0.7 0.8 0.7 0.8 4.4 4.9 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8
P3 1.2 1.1 0.7 0.3 0.7 0.8 4.4 4.1 1.1 1.2 1.3 1.4 7.5 7.6 7.7 7.8
P5 2 2 1 1 1 1 5 5 1 2 3 4 5 6 7 8
Vehicle Number
Sequence Number

Figure 13. Coordinate transfer after RM.

The value mutation and sorting mutation functions described were executed after each iteration of individual location coordinate updates and before the calculation of the objective function. The results of RM directly updated the individual location coordinates, serving as the basis for calculating the objective function and planning the next generation of individual distribution information. Because

RM converted the distribution genes represented by the location coordinates into discrete integer variables, the distribution of independent variables in the objective function was discrete. This eliminated small changes in continuous space among individuals, thus avoiding the occurrence of MVC.

3.4. MDO process

Algorithm: Discrete optimization algorithm pseudocode

```
initialize system parameters
establish feasible reference
while (loop1 <= SetNum1)
    randomly generate partitioned agents in the optimization space
    record partition agents information
    add the priori agents to the population
    calculate the sample size for each partition
    for $Np <= Np_{max}$ do
        randomly generate base agents within the $Np$-th partition
        record base agents information
    end for
    calculation of base generation derived sampling size
    while (loop2 <= SetNum2)
        for $Nb <= Nb_{max}$ do
            generate sub-generation agents around the $Nb$-th base generation
            record sub-generational agent information
        end for
        for $Np <= Np_{max}$ do
            update partition agents and calculate sub-generation agent parameters
        end for
            revise sub-generational agents with updated sampling information
            survival of the fittest, iterative transformation
            loop2++
    end while
    loop1++
end while
```

The optimization process followed a systematic series of steps. Initially, the population was initialized by randomly generating sampling points within the feasible region, acting as the initial optimization agents. These agents, defined by their location coordinates or distribution genes, served as decision variables. An RM was then applied to transform continuous real coordinates into discrete integer coordinates. The objective function, representing transportation costs for various distribution schemes, was calculated based on the updated coordinates. Subsequently, a population of offspring agents was generated around each individual, with a focus on evolution factors and variation in the objective function. Descendant agents underwent a similar mutation process, and winners were selected based on their objective function values. The process was iterated through offspring reproduction until a specified condition was met. The dominant agent, representing the optimal individual, was recorded, and the procedure continued until the exit task was executed, consequently outputting the optimal distribution solution and concluding the program.

Figure 14 shows the flowchart of the discrete optimization algorithm that has been designed for the logistics distribution cost optimization problem. Its detailed functional process is described as follows:

Step 1. The population was initialized by randomly generating sampling points in the feasible region as the initial optimization agents. The agents' location coordinates were the decision variables for optimization, and they were composed of distribution genes. If there was an established dominant population, it was added to the initial population of agents to form the initial population.

Step 2. The RM of the agent's location coordinates was performed. The continuous real coordinates were transformed into discrete integer coordinates, and the updated location coordinates were used as the coordinates of the current sampling point to perform the objective function calculation; then, the transportation costs of different distribution schemes were obtained.

Step 3. The objective function value, the amount of objective function value variation, the distribution range, and the number of offspring populations generated from the current individual were calculated by combining the evolution factor (i.e., learning rate).

Step 4. With each agent individual as the center, we generated a population of uniform sampling points according to the range and number of offspring as a population of offspring optimization agents. At the same time, the current agent also transformed itself into a descendant agent and participated in the analysis and operation of the descendants. In this step, the objective function was not calculated.

Step 5. The position coordinates of the descendant agents were rationally mutated. The continuous real coordinates were transformed into discrete integer coordinates, the updated position coordinates were used as the coordinates of the sampling points of the descendants, and the calculation of the objective function of the descendants was performed to obtain the transportation cost of the derived distribution scheme.

Step 6. The winners were selected based on the objective function value of the descendant individuals, and the winning individuals were retained as the current generation of optimization agents to participate in the subsequent processing.

Step 7. We determined whether to end offspring reproduction. If yes, the next step was performed; if not, it looped back to Step 3 for the next-generation reproduction process and new distribution schemes were derived.

Step 8. The information of the current dominant agent individual was recorded, especially focusing on the optimal individual that was explored.

Step 9. We determined whether to end the procedure. If yes, the next processing session was performed; if not, it looped back to Step 1 and the superiority-seeking exploration was continued.

Step 10. The exit task was executed, the optimal distribution solution and cost results were output, and the program was ended.

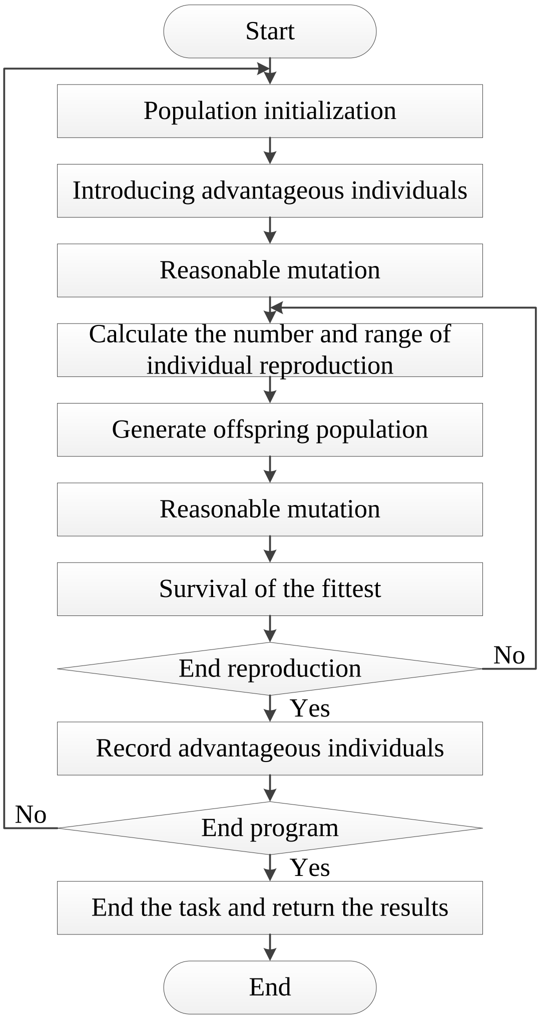

Start
Population initialization
Introducing advantageous individuals
Reasonable mutation
Calculate the number and range of individual reproduction
Generate offspring population
Reasonable mutation
Survival of the fittest
End reproduction
No
Yes
Record advantageous individuals
End program
No
Yes
End the task and return the results
End

Figure 14. Flowchart of the discrete optimization algorithm.

4. Experiments and analysis

Similar to classical swarm intelligence optimization algorithms, the MO algorithm is also a heuristic swarm optimization algorithm. This type of algorithm is suitable for objective functions but not suitable for mathematical optimization methods, and it is difficult to establish mathematical theoretical analysis for such algorithms. Swarm intelligence optimization algorithms are generally tested by using benchmark functions to demonstrate the improvements of the new algorithm and thus objectively analyze the effectiveness of the algorithm. We first validated the superior performance of the new algorithm by using six benchmark functions, and then we examined the effectiveness of the new algorithm in terms of solving discrete optimization problems in logistics optimization cases.

4.1. Experimental tools

The continuous optimization test was conducted to assess the initial performance of the optimization algorithm by using benchmark functions, such as Griewank, Needle, Rastrigin, Rosenbrock, Schaffer, and Shubert. This test compared the classical GA, PSO, and MCO (MO with continuous optimization) to provide a more intuitive reflection of the MO algorithm's optimization performance. The parameters for each algorithm were set as listed in Table 2.

Table 2. Parameters for three algorithms.

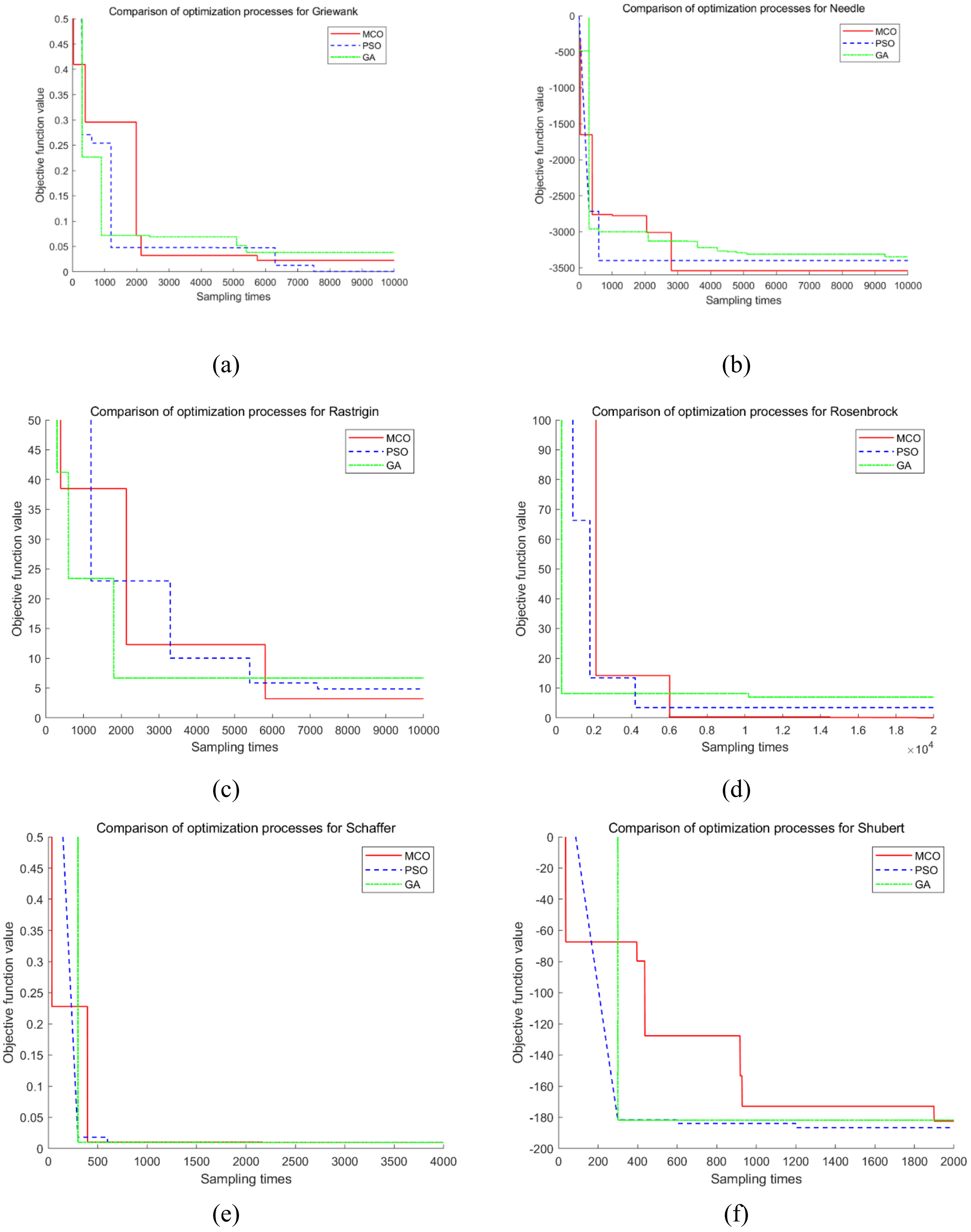
Algorithms	Parameters	Value
GA	Number of chromosomes	100
GA	Iteration evolution upper limit	5000
GA	Probability of variation	5%
GA	Termination condition for invalid evolution	20
GA	Encoding accuracy of a single variable	15 bits
PSO	Number of population individuals	300
PSO	Maximum iterations	1200
PSO	Detection interval during optimization	30
PSO	Termination condition for invalid progress	0.001
PSO	Optimal weight coefficient for the individual	1.5
PSO	Optimal weight coefficient for the whole	0.5
MCO	Initial number of population individuals	36
MCO	Maximum outer cycles	20
MCO	Maximum inner cycles	2
MCO	Evolution factor (learning rate)	0.5
MCO	Elimination rate	0.7

4.2. Continuous optimization

Continuous optimization tests were conducted for various benchmark functions; the representative optimization processes were as illustrated in Figure 15(a)–(f).

The horizontal axis in Figure 15 represents the number of samples, that is, the cumulative number of samples. This logical quantity analysis helped us to eliminate the influence of the computer environment on the algorithm speed. The vertical axis in Figure 15 represents the objective function. For example, in minimax optimization, a faster decrease in the curve indicated a faster optimization search.

Analysis of the optimization results revealed that the GA demonstrated good exploration ability by quickly capturing better values in the early stages of optimization. However, in the later stages of the optimization search, the objective function value curve flattened, indicating insufficient exploration ability. PSO also quickly captured better values early on but sometimes experienced prolonged stagnation in the intermediate phase between 1500 and 6000 samples (Figure 15(a)). The GA and MCO both showed multiple instances of new optimal values and outperformed PSO before the 6000th sample. This was because PSO got temporarily trapped in a local optimal region, resulting in a slower overall optimization process. The MCO algorithm's overall optimization process was faster, allowing it to jump out of local optima earlier and speed up global optimum convergence. Particularly for the deceptive Needle problem, MCO jumped out of local optima and converged to the global optimum around the 3000th sample, while the GA and PSO remained stuck in local optima even after 10,000 samples.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 15. Comparison of nominal optimization processes.
(The tests in (a), (b), (c), (d), (e) and (f) used benchmark functions Griewank, Needle-in-a-haystack, Rastrigin, Rosenbrock, Schaffer and Shubert respectively.)

Table 3. Statistics of optimization test results.

NO.	Griewank			Needle			Rastrigin		
	GA	PSO	MCO	GA	PSO	MCO	GA	PSO	MCO
1	0.0087	0.0000	0.0000	-3546.4099	-3596.8927	-3597.5196	0.2212	0.0000	0.0004
2	0.0044	0.0000	0.0000	-3599.4862	-3599.9991	-3595.2110	1.1718	0.0000	0.0052
3	0.0058	0.0000	0.0001	-3599.8834	-3401.2250	-3592.3091	0.1541	0.0000	0.0448
4	0.0085	0.0000	0.0001	-3576.4285	-3600.0000	-3596.2463	1.8767	0.0000	0.0000
5	0.0026	0.0000	0.0001	-3598.5653	-3600.0000	-3594.6778	1.5559	0.0000	0.0002
6	0.0004	0.0000	0.0001	-3595.4264	-3600.0000	-3598.5334	2.4279	0.0000	0.0004
7	0.0016	0.0000	0.0002	-3599.7370	-3600.0000	-3599.4721	1.2546	0.0000	0.0001
8	0.0048	0.0000	0.0001	-3597.8252	-3600.0000	-3599.8120	1.1212	0.0000	0.0024
9	0.0010	0.0000	0.0001	-3597.8611	-3600.0000	-3598.0424	2.3655	0.0000	0.0337
10	0.0018	0.0000	0.0001	-3563.4747	-3401.2250	-3594.5222	0.0495	0.0000	0.0567
11	0.0001	0.0000	0.0001	-3542.4302	-3600.0000	-3599.5722	0.1541	0.0000	0.0078
12	0.0042	0.0000	0.0001	-3587.6483	-3600.0000	-3593.0288	1.3851	0.0000	0.0049
13	0.0040	0.0000	0.0000	-3599.5134	-3401.2250	-3599.5746	1.3290	0.0000	0.0515
14	0.0086	0.0000	0.0007	-3598.2640	-3600.0000	-3598.1535	1.2866	0.0000	0.0026
15	0.0079	0.0000	0.0000	-3591.4437	-3401.2250	-3599.4319	0.8335	0.0000	0.0055
16	0.0004	0.0000	0.0001	-3599.0221	-3600.0000	-3597.3215	0.5085	0.0000	0.0042
17	0.0001	0.0000	0.0000	-3586.5276	-3600.0000	-3599.7086	0.8335	0.0000	0.0039
18	0.0012	0.0000	0.0000	-3545.4849	-3599.9997	-3598.4448	1.0196	0.0000	0.0120
19	0.0077	0.0000	0.0002	-3583.2710	-3600.0000	-3595.1445	2.1842	0.0000	0.0040
20	0.0083	0.0000	0.0005	-3597.8120	-3600.0000	-3599.9242	0.1541	0.0000	0.0063
Mean	0.0041	0.0000	0.0001	-3585.3258	-3560.0896	-3597.3325	1.0943	0.0000	0.0123
SD	0.0032	0.0000	0.0002	19.3260	79.4351	2.3516	0.7282	0.0000	0.0178
Max	0.0087	0.0000	0.0007	-3542.4302	-3401.2250	-3592.3091	2.4279	0.0000	0.0567
Min	0.0001	0.0000	0.0000	-3599.8834	-3600.0000	-3599.9242	0.0495	0.0000	0.0000

Table 3 presents the statistics resulting from multiple optimization tests for the first three nominal functions. In most cases, PSO demonstrated good convergence ability, achieving global optimal results with high accuracy for Griewank and Rastrigin, while the GA and MCO showed relatively lower accuracy. However, regarding the minimum statistics, MCO and PSO achieved very similar results in some cases. Conversely, for the Needle problem, MCO exhibited better accuracy with the smallest mean (–3597.3325) and the smallest standard deviation (2.3516) of optimization results, indicating good and stable overall optimization performance with superior global search capability. The maximum optimization result (–3592.3091) suggested that the algorithm captured the global optimal region consistently. In contrast, the mean (–3560.0896) and standard deviation (79.4351) of the PSO optimization search results in the group test indicated a deviation from the global optimal region. However, its minimum value for the optimization search result (–3600.0000) reached the global optimum. This suggested that strong convergence ability may temporarily affect optimization accuracy when the algorithm is trapped in local optima, but higher accuracy results quickly emerge once the local optima are escaped.

In summary, the group intelligence optimization algorithm was influenced by group evolution, introducing uncertainty into the optimization process and results. However, the statistical results showed that MCO exhibited minimal standard deviation and relatively good stability. Its minimum optimization results nearly approached the global optimum, and the optimization convergence process demonstrated a faster exploration speed in the optimization region. These characteristics highlighted that MCO effectively balanced global exploration and local exploitation, providing a solid foundation for its application to high-dimensional discrete optimization in logistics and distribution.

4.3. Discrete optimization

This study examined the abstract model for logistics and distribution, expanding the distribution sites from 11 sites (including freight centers) in Table 4 to 21 sites. The spacing between sites was assumed to be uniform, the distance from the freight center increased sequentially, and the distribution cost was directly proportional to the transportation distance. Without sacrificing generality, the problem model effectively captured the intricacies of high-dimensional discrete optimization for multivehicle, multisite distribution problems. The logical representation of the correlation between site distances and distribution costs encapsulated the fundamental characteristics of real-world distribution challenges. Moreover, the parameters used in this test remained consistent with those employed in the continuous optimization test in the evaluation of the algorithm's generalization capability.

Table 4. Sample table of intersite distribution costs.

Site	0#	1#	2#	3#	4#	5#	6#	7#	8#	9#	10#
0	0	10	20	30	40	50	60	70	80	90	100
1	10	0	10	20	30	40	50	60	70	80	90
2	20	10	0	10	20	30	40	50	60	70	80
3	30	20	10	0	10	20	30	40	50	60	70
4	40	30	20	10	0	10	20	30	40	50	60
5	50	40	30	20	10	0	10	20	30	40	50
6	60	50	40	30	20	10	0	10	20	30	40
7	70	60	50	40	30	20	10	0	10	20	30
8	80	70	60	50	40	30	20	10	0	10	20
9	90	80	70	60	50	40	30	20	10	0	10
10	100	90	80	70	60	50	40	30	20	10	0

The optimization process for the logistics and distribution scheme is illustrated in Figure 16, offering a representative comparison of the optimization and convergence processes. The red and blue solid lines in the figure represent the optimization curves for MDO (MO applied to discrete optimization) and D-PSO (PSO with discrete optimization), incorporating an RM mechanism for sampling coordinates. The red and blue dashed lines represent the optimization curves for MCO and C-PSO (PSO with continuous optimization), lacking an RM mechanism for sampling coordinates. In terms of the final optimization results, the addition of an RM mechanism significantly enhanced the accuracy, with MDO demonstrating more pronounced effects.

Considering the convergence speed of the optimal result, regardless of the presence of the RM mechanism, PSO exhibited notable rapidity in the early stages of optimization. After approximately 7000 cumulative samples, the PSO optimal value approached nearly 500. However, the PSO convergence was relatively weak in the later stages, particularly for C-PSO without RM, where the final optimization result did not surpass 400. The optimization speed of MCO without RM was relatively slow due to the algorithm parameter settings favoring exploration. Without the RM of sampling coordinates, the objective function of the optimization space was presented as a continuous distribution with MVCs, imposing a burden on exploration.

Upon transforming MCO into MDO through the introduction of an RM mechanism, the undesirable virtual changes were eliminated. The objective function of the optimization space could then be represented as a discrete distribution, significantly improving the speed and accuracy of the optimal result convergence. As indicated by the red solid line in Figure 16, the convergence speed of MDO in the early stages was not as rapid as that of D-PSO. However, it caught up before reaching about 20,000 samples through a more extended period of effective breakthrough. After reaching approximately 80,000 samples, MDO surpassed D-PSO's breakout speed and continued toward the global optimal result, eventually converging to a distribution cost of 200.

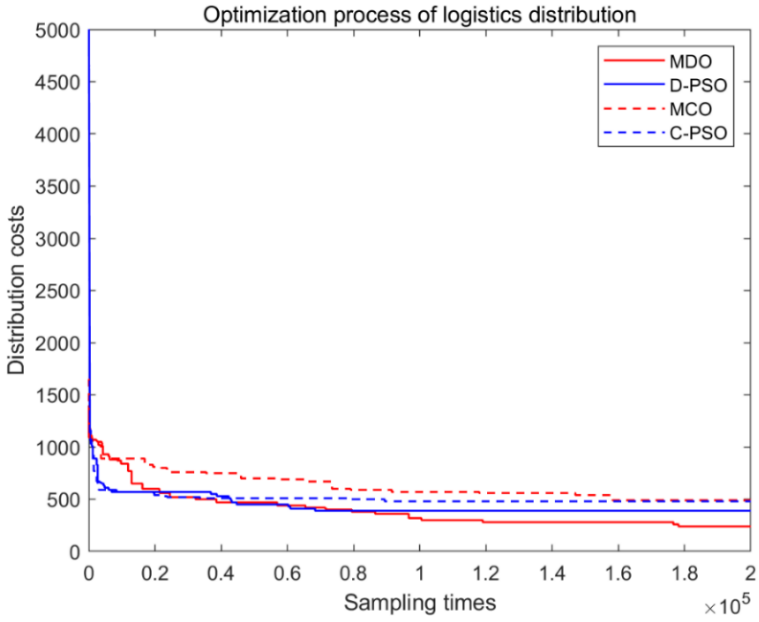


Figure 16. Comparison of optimization process for logistics delivery schemes.

Given the inherent uncertainty in the results of heuristic optimization algorithms, we employed a testing approach involving multiple trials and multiple iterations per trial. The distribution cost results obtained from multiple optimization searches of D-PSO are presented in Table 5. The top row of the table indicates the group number, ranging from Sets 1 to 10. Each group consisted of 10 tests, identified from 1 to 10 in the leftmost column. The table encompassed 100 tests across these 10 trials. Upon examining the statistical outcomes, it was observed that D-PSO identified the global optimal distribution cost of 200 only after seven out of 10 rounds of tests. The average results for each round ranged between 352 and 456.

Table 5. Records and statistics of multiple optimization results for D-PSO.

Site	0#	1#	2#	3#	4#	5#	6#	7#	8#	9#
1	490	480	510	380	510	390	450	530	440	370
2	410	290	520	270	240	390	200	400	340	450
3	380	470	390	340	330	240	390	540	390	490
4	280	460	260	540	410	520	300	410	370	440
5	550	490	580	290	430	370	400	260	390	390
6	470	420	400	330	290	530	440	450	360	540
7	370	380	410	250	440	390	310	620	420	480
8	450	440	430	570	290	380	410	340	520	400
9	300	500	550	430	380	510	380	470	450	560
10	240	460	390	400	390	390	240	310	560	440
Mean	394	439	444	380	371	411	352	433	424	456
SD	94.4	59.9	91	102.9	78.4	83.4	80.8	106.2	67.1	59.2
Max	550	500	580	570	510	530	450	620	560	560
Min	240	290	260	250	240	240	200	260	340	370

Table 6. Records and statistics of multiple optimization results for MDO.

Site	0#	1#	2#	3#	4#	5#	6#	7#	8#	9#
1	210	400	240	220	200	260	200	370	550	510
2	370	240	200	250	290	380	200	430	450	290
3	200	290	450	200	200	260	430	200	260	220
4	200	210	200	200	200	220	200	220	220	370
5	200	440	500	200	200	200	200	350	250	200
6	200	370	200	200	270	200	400	330	320	230
7	200	470	480	420	200	260	260	200	280	270
8	240	340	200	200	370	380	200	200	200	200
9	370	360	280	350	390	220	450	350	280	200
10	200	380	230	280	200	440	200	480	420	200
Mean	239	350	298	252	252	282	274	313	323	269
SD	66.6	78.6	119.9	72.9	71.4	81.7	102.1	97.4	107.6	95.8
Max	370	470	500	420	390	440	450	480	550	510
Min	240	290	260	250	240	240	200	260	340	370

Table 6 presents the results of distribution costs obtained from multiple optimization searches conducted by MDO. With the exception of the second round, all other rounds resulted in a global optimal distribution cost of 200, demonstrating the high accuracy of MDO. Compared with the optimization results of D-PSO, the average value of optimal distribution costs obtained by MDO ranged from 239 to 350, which was smaller than the average result for D-PSO. This suggested that MDO demonstrated better overall performance. The standard deviation for D-PSO ranged from 59.2 to 106.2, while that for MDO ranged from 66.6 to 119.9, indicating comparable stability between the two algorithms. The maximum optimization results for D-PSO were between 450 and 620, whereas those for MDO were between 370 and 550. This suggests that both algorithms encountered local optima, impacting their stability to varying extents. The minimum optimization results for D-PSO were between 200 and 370, with 10% of the optimal results being ≤ 200. In contrast, the corresponding results for MDO were between 200 and 210, with 90% of the optimal results being ≤ 200. This demonstrated a significant improvement in algorithm performance with only 10 instances of optimization result statistics, with MDO showing a more pronounced improvement, offering approximately a 90% chance of finding the global optimal result.

5. Conclusions

The proposed MO algorithm comprises two branches: MCO and MDO, tailored for different types of optimization problems.

First, addressing continuous optimization, we introduced the MCO algorithm. This algorithm leverages the heuristic changes in successive sampling information through the dynamic division of multiple regions, adapting to various types of changes in sampling density within these regions. The optimization mode was designed to ensure a faster convergence speed while preserving exploration capability in multiple global dominant regions, effectively preventing the optimization process from being hampered by local optima. As a result, the MCO algorithm demonstrated superior performance in terms of overall optimization speed and accuracy.

Second, for discrete optimization problems, we investigated the MVC issue that tends to arise when applying continuous optimization algorithms. To address this, we proposed the MDO algorithm. MDO incorporates the RM mechanism of agent coordinates to confine continuously distributed optimization sampling points to the discrete decision variable space. This mitigates the issue of excessively dense sampling in the optimization space, effectively resolving the problem of MVC. Application to logistics and distribution optimization problems indicated a significant improvement in optimization performance as a result of transforming continuous optimization algorithms into discrete optimization algorithms by using the RM mechanism.

Finally, we thoroughly validated the superiority of MDO in the task of optimizing exploration by conducting statistical tests on uncertainty in the heuristic optimization algorithm results. The statistical scheme involved multiple rounds of tests, revealing that the MDO algorithm's enhancement was more pronounced after multiple optimization searches. This suggests that MDO maximized the benefits of a large sample number for comprehensive exploration, further improving the algorithm accuracy and stability. For more complex practical optimization problems that involve conducting relatively large numbers of optimization searches an require improved result reliability, algorithms with superior exploration performance, such as MO, should be preferred.

Use of AI tools declaration

The authors declare that they have not used artificial intelligence tools in the creation of this article.

Acknowledgements

This study was supported by the Taizhou Industrial Science and Technology Plan Project (Grant 21gyb23), the Taizhou Science and Technology Bureau Innovation Voucher Service Project (Grant CXQ-2021-023), and the Taizhou University Scientific Research Fund.

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. X. Zhu, N. Liu, Y. Shi, Artificial intelligence technology in modern logistics system, *Int. J. Technol. Policy Manage.*, **22** (2022), 66–81. https://doi.org/10.1504/IJTPM.2022.122537
2. F. Qiu, Focus on the thinking of logistics operations developing to the digital and intelligent supply chain, *Hoisting Conveying Mach.*, (2021), 22–23.
3. W. Qin, X. Qi, Evaluation of green logistics efficiency in northwest China, *Sustainability*, **14** (2022), 6848–6848. https://doi.org/10.3390/su14116848
4. Z. Yang, Research on the construction of paper industry information logistics system under the background of green and recyclable, *Paper Sci. Technol.*, **40** (2021), 77–79. https://doi.org/10.19696/j.issn1671-4571.2021.4.016
5. S. Zhao, Q. Zhang, Z. Peng, X. Lu, Personalized manufacturing service composition recommendation: combining combinatorial optimization and collaborative filtering, *J. Comb. Optim.*, **40** (2020), 733–756. https://doi.org/10.1007/s10878-020-00613-0
6. H. Zhang, H. Wang, Y. Wang, D. Hong, Uncertainty modeling and optimization method for overall design of flight vehicle, *J. Astronaut.*, **44** (2023), 486–495. https://doi.org/10.3873/j.issn.1000-1328.2023.04.003
7. Y. Zhang, L. Liu, T. Wang, J. Guo, Y. Han, Error analysis and optimization algorithm of focal shift on mode decomposition for few-mode fiber beam, *Optoelectron. Lett.*, **17** (2021), 418–421. https://doi.org/10.1007/s11801-021-0147-x
8. Y. Hua, Y. Liu, W. Pan, X. Diao, H. Zhu, Multi-objective optimization design of bearingless permanent magnet synchronous motor using improved particle swarm optimization algorithm, *Chin. Soc. Elec. Eng.*, **43** (2023), 4443–4451. https://doi.org/10.13334/j.0258-8013.pcsee.220039
9. D. Wang, S. Chen, Spatial difference and distribution dynamic evolution of high-quality development of logistics industry in China, *Stat. Decis.*, **38** (2022), 57–62. https://doi.org/10.13546/j.cnki.tjyjc.2022.09.011
10. J. Wang, K. Zhou, Analysis of the relationship between the distribution of city size and the allocation efficiency of logistics industry, *J. Commer. Econ.*, **22** (2023), 92–96. https://doi.org/10.3969/j.issn.1002-5863.2023.22.021
11. K. Shang, X. Wei, The effect of integration development of logistics industry and information industry on its energy intensity, *J. Henan Univ.*, **63** (2023), 19–23,152. https://doi.org/10.15991/j.cnki.411028.2023.03.021
12. C. Zhang, J. Liu, Multi-box container loading problem based on hybrid genetic algorithm, *J. Beijing Univ. Aeronaut. Astronaut.*, **48** (2022), 747–755. https://doi.org/10.13700/j.bh.1001-5965.2020.0665
13. G. Zhao, Z. Qin, J. Li, Optimization algorithm and implementation of dispatched vehicles between several flights in condition of flights delay, *J. Chongqing Jiaotong Univ.*, **39** (2020), 5–9,17. https://doi.org/10.3969/j.issn.1674-0696.2020.10.02
14. Y. Liu, H. Chen, Distribution path planning and charging strategy for pure electric vehicles with load constraint, *J. Comput. Appl.*, **40** (2020), 2831–2837. https://doi.org/10.11772/j.issn.1001-9081.2020020157

15. H. Lu, K. Zhao, Optimization of the multi-containers loading problems based on complex constraints, *J. Wuhan Univ. Technol. (Transp. Sci. Eng.)*, **40** (2016), 1058–1062. https://doi.org/10.3963/j.issn.2095-3844.2016.06.023
16. N. Wu, H. Dai, J. Li, Q. Jiang, Multi-objective optimization of cold chain logistics distribution path considering time tolerance, *J. Transp. Syst. Eng. Inf. Technol.*, **23** (2023), 275–284. https://doi.org/10.16097/j.cnki.1009-6744.2023.02.029
17. H. Lu, Y. Wang, Research on preventive maintenance strategy of port machinery equipment considering reliability and economy, *Modern Manuf. Eng.*, (2022), 116–122,115. https://doi.org/10.16731/j.cnki.1671-3133.2022.08.018
18. M. Zhang, Y. An, N. Pan, Y. Sun, J. Bao, H. Gao, et al., Heterogeneous vehicle scheduling oriented to urban city supermarket logistics distribution, *J. Kunming Univ. Sci. Technol.*, **47** (2022), 154–162. https://doi.org/10.16112/j.cnki.53-1223/n.2022.06.482
19. J. Li, Study on the development path of green logistics from the perspective of supply chain, *Logist. Sci.-Tech.*, **46** (2023), 133–135,139. https://doi.org/10.13714/j.cnki.1002-3100.2023.14.036
20. Y. Wang, Y. Wei, Q. Jiang, M. Xu, Study on the optimization method of three-dimensional loading logistics distribution with time windows, *Oper. Res. Manage. Sci.*, **31** (2022), 111–119. https://doi.org/10.12005/orms.2022.0395
21. J. Huang, Research on cross-border e-commerce logistics distribution optimization based on multi-objective optimization and chicken flock algorithm, *Econ. Res. Guide*, (2021), 34–37. https://doi.org/10.3969/j.issn.1673-291X.2021.03.012
22. C. Wu, J. Yang, Vehicle routing problem of logistics distribution based on improved particle swarm optimization algorithm, *Comput. Eng. Appl.*, **51** (2015), 259–262. https://doi.org/10.3778/j.issn.1002-8331.1409-0200
23. J. Lu, Scheduling distribution vehicles in internet of things based on perturbation contraction particle swarm optimization, *J. Highway Transp. Res. Dev.*, **37** (2020), 111–117. https://doi.org/10.3969/j.issn.1002-0268.2020.04.015
24. W. Fei, C. Liu, S. Hu, Research on swarm intelligence optimization algorithm, *J. China Univ. Posts Telecommun.*, **27** (2020), 1–20. https://doi.org/10.19682/j.cnki.1005-8885.2020.0012
25. Z. Wang, Y. Deng, Optimizing financial engineering time indicator using bionics computation algorithm and neural network deep learning, *Comput. Econ.*, **59** (2022), 1755–1772. https://doi.org/10.1007/s10614-022-10253-7
26. P. Trojovsky, M. Dehghani, A new optimization algorithm based on mimicking the voting process for leader selection, *PeerJ Comput. Sci.*, **8** (2022), 1–40. https://doi.org/10.7717/peerj-cs.976
27. S. Bhatti, V. Tayal, P. Gulia, Swarm intelligence, *Int. J. Innovative Res. Technol.*, **1** (2015), 210–214.
28. R. Qi, Z. Wang, S. Li, A parallel genetic algorithm based on spark for pairwise test suite generation, *J. Comput. Sci. Technol.*, **31** (2016), 417–427. https://doi.org/10.1007/s11390-016-1635-5
29. W. Zhou, S. Li, G. Ma, X. Chang, X. Ma, C. Zhang, Parameters inversion of high central core rockfill dams based on a novel genetic algorithm, *Sci. China Technol. Sci.*, **59** (2016), 783–794. https://doi.org/10.1007/s11431-016-6017-2
30. Ankita, S. K. Sahana, Ba-PSO: A Balanced PSO to solve multi-objective grid scheduling problem, *Appl. Intell.*, **52** (2022), 4015–4027. https://doi.org/10.1007/s10489-021-02625-7
31. P. Kumari, S. K. Sahana, PSO-DQ: An improved routing protocol based on PSO using dynamic queue mechanism for MANETs, *J. Inf. Sci. Eng.*, **38** (2022), 41–56. https://doi.org/10.6688/JISE.202201_38(1).0003

32. Y. Xiang, X. Yang, H. Huang, J. Wang, Balancing constraints and objectives by considering problem types in constrained multiobjective optimization, *IEEE Trans. Cybern.*, **53** (2023), 88–101. https://doi.org/10.1109/TCYB.2021.3089633

33. Y. Sui, Z. Li, H. Li, G. Chen, Continuous solution method for 0-1 programming based on the sinusoidal smooth polish function, *Oper. Res. Trans.*, **21** (2017), 35–44. https://doi.org/10.15960/j.cnki.issn.1007-6093.2017.03.004