J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 9157--9179

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Numerical Solution and Accuracy Improvement of Garment Size Matching Problem Based on Optimization Algorithm

Mengmeng Hou[1,✉]

[1] *Department of Fashion and Apparel Design, Zhengzhou Academy of Fine Arts, Zhengzhou, Henan, 451450, China*

ABSTRACT

In the garment production industry, garment cutting size matching plan is an important step in the process, which plays a decisive role in production management and cost control. In this paper, we first model the size matching problem of garment cutting, then use the improved fast particle swarm algorithm (APSO) to optimize the multi-objective optimization solution, and finally verify the performance of the APSO algorithm and the actual effect of garment size matching with cases. Comparing the test results of APSO, PSO and LDWPSO algorithms in the six test functions of Griewank, Ackle, Levy, Rastrign, Schwefel and Sphere, it can be seen that: with the improvement of the problem dimensions, the APSO algorithm used in this paper can still maintain a better optimization accuracy, and the optimization accuracy and stability are significantly improved compared with the PSO and the LDWPSO algorithms. LDWPSO algorithms. In the actual case, the APSO algorithm is more reasonable in the size combination and the number of layers of fabric, for four different types of apparel orders have obtained a superior optimal solution set, cutting production error is far less than the enterprise requirements. At the same time, compared with other optimization methods, the APSO algorithm has better optimization accuracy and solving efficiency, and can obtain a more superior cutting and bed splitting scheme. The algorithm proposed in this paper can effectively optimize the cutting size matching process, reduce fabric waste and production equipment investment, and has good application value and reference significance.

Keywords: multi-objective optimization, APSO algorithm, garment size matching, accuracy improvement

1. Introduction

Optimization algorithm refers to the process of increasing the efficiency of an algorithm or decreasing the consumption of resources by improving an existing algorithm or proposing a new algorithm when solving a problem. Comparative analysis of optimization algorithms refers to comparing and

✉ Corresponding author.
E-mail address: 13703848103@163.com (M. Hou).
Received 20 February 2024; Revised 25 April 2024; Accepted 20 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-502

evaluating several different optimization algorithms to determine which algorithm is most suitable for solving some specific problems [1-4]. Different optimization algorithms are suitable for different types of problems, and the selection of an appropriate algorithm depends on the characteristics and requirements of the problem. In practical applications, the effectiveness of optimization algorithms usually needs to be adjusted and verified according to specific situations. It can play an important role in the problems related to clothing size matching [5-8].

Clothing size refers to the standard of measuring the size of clothing, which is crucial for consumers to buy the right size of clothing. Clothing size usually includes the length, bust, shoulder width, sleeve length, waist, hip, and other dimensions of clothing. The measurement standards of these dimensions are based on the actual size and proportion of the human body [9-12]. There are some differences in human body sizes and proportions in various countries and regions, thus leading to different clothing sizing standards in different countries and regions. Consumers should be aware of their actual sizes when purchasing clothing [13-16]. Secondly, consumers need to choose the appropriate clothing size according to their actual size. When choosing clothing size, consumers need to take into account the shape of the clothing, fabric elasticity and other factors to ensure that the purchased clothing size can fit comfortably [17-20]. As consumers' demand for personalization continues to rise, the accurate matching of clothing sizes has become a key factor affecting consumers' purchasing experience. Consumers need to choose the appropriate clothing size according to their own actual situation to ensure that the clothing is comfortable to wear and meets their own image needs [21-24].

In this paper, a multi-objective optimization model for garment cutting size matching is constructed based on the Improved Fast Particle Swarm Algorithm (APSO), and the validity of the constructed model is verified through the accuracy evaluation of the APSO algorithm and case applications. In the algorithm experiments, the APSO algorithm is tested by six functions, Griewank, Ackle, Levy, Rastrign, Schwefel, and Sphere, and the PSO and LDWPSO algorithms are compared in terms of optimization results. In the case applications, the APSO algorithm is compared with SVM-PSO, PSO, GA-PSO, LINGO, LDWPSO algorithms to verify the significant advantages of the proposed algorithm over traditional algorithms and optimization software tools in terms of solution accuracy and efficiency.

2. A multi-objective optimization model for garment size matching based on APSO

2.1. Formulation of the problem

With the price of raw materials and labor wages rising, the cost pressure on the apparel industry is increasing, and if enterprises want to stand in the apparel industry, they must consider how to save production costs. In the garment manufacturing process, the most decisive role in saving raw materials is the cutting process, so if the cutting process can be optimized, it will make a significant contribution to the cost reduction of enterprises. Therefore, the optimization of cut-to-size (CSM) is an issue worth studying. Firstly, we have to consider the raw material cost, i.e., the distribution solution given has to be practical and feasible, how to make the distribution result in the lowest fabric cost and the highest overall fabric utilization. Secondly, we have to consider the labor cost, and here we mainly focus on the labor cost of the cutting shop, and the labor wage billing of the cutting shop is based on hourly wage, so we hope that the allocation scheme can make the cutting time reach the shortest so as to achieve the lowest labor cost and also improve the production efficiency. Therefore, in actual production, in order to maximize the savings of fabric, the production of mark will be different sizes of the same part of the clothes cut pieces, according to a certain number of ratios in a mark drawing. The use of professional CAD software for nesting, so that the nesting map is the most compact, the highest utilization rate, while the section length of the mark map will not exceed the maximum length of the cutting machine. A mark bank is created using the marks obtained after nesting, and then the mark bank is used to allocate orders based on constraints such as the number of orders, thereby realizing the goals identified in the demand analysis.

2.2. Modeling the size matching problem for multi-objective optimization of garment cutting

Most of the practical problems have multiple objectives and need to satisfy the given conditions at the same time, i.e., in the same problem model, there are several nonlinear objectives at the same time, and these objective functions need to be optimized at the same time, and this kind of problem is known as a multi-objective optimization problem. For the sake of generality, assuming that there are n decision variable and ρ objectives, the mathematical model of the multi-objective optimization problem can be described as equation (1):

$$\begin{gathered} \min y = F(x) = \left(f_1(x), f_2(x), \cdots, f_p(x)\right) \\ subjecttog_j(x) \le 0, j = 1,2,\ldots,J \\ h_k(x) = 0, k = 1,2,\ldots,K \\ x \in D \end{gathered} \tag{1}$$

where $x \in D \subset R^n$ is the n-dimensional decision variable, $x = \left(x_1, x_{2,\ldots,}, x_{i,\ldots,}, x_n\right)$, x_i denotes the i th decision variable, and D is the n-dimensional decision space. $y \in S \subset \square^{\rho}$ is the ρ-dimensional solution space. $F(x)$ defines ρ objective component functions mapping from decision space to solution space. Where $f_i(x)$ is the i th objective component of the objective function. $g_f(x) \le 0\left(j = 1,2,\ldots,J\right)$ defines the J inequality constraints, and $h_k(x) = 0\left(k = 1,2,\ldots,K\right)$ defines the K equality constraints. $\min$ or $\max$ represents the optimization model of the multi-objective problem.

Cut-to-size matching (CSM) system is to develop a CSM scheme with the constraints of the number of order sizes, the number of marks, and the number of layers of fabric on the cutting bed, and with the objectives of minimizing the cost of materials, maximizing the utilization rate, and minimizing the time spent in the cutting process. At the same time, the problem under study has multiple objectives and is a multi-objective optimization problem, so equation (1) is applied to develop a mathematical model.

In order to better describe the modeling process of multi-objective optimization cutting size matching, the variables used are defined as follows:

m : The number of sizes in the order by garment style size.

d_j : Number of orders in the j rd size $(j = 1,2,\ldots,m)$.

φ_j : Number of pieces allowed for error for the j th size $\left(\varphi_j \ge 0\right)$.

n : Number of Marks in the Mark Bank generated by the Mark Generation System.

a_{ij} : Number of matching pieces for the j th size in the i th mark $(i = 1,2,\ldots,n)$.

l_i : Length of the fabric section used for the i th mark.

u_i : The utilization rate of the material used in the i nd Mark.

p_i : i th Mark used size fabric price.

t_i : Mark i Cutting time.

x_i : i th mark number of layers to be distributed.

c_i : i th mark number of beds distributed.

kp_j : The actual number of sizes allocated for the j nd size.

x : Distribution program, $x = \left(x_1, x_2, \ldots, x_n\right)^T$.

v_t : Fabric pulling speed of the fabric pulling machine.

$L_{\max}$: Maximum number of mark layers to be cut on the cutting bed.

$L_{\min}$: Minimum number of mark layers to be cut on the cutting bed.

2.2.1. Identification of decision variables. The final cropping scheme needed is determined by x and if $x_P > 0$, then mark i will be used. The value of x_i will affect the quality of the final allocation scheme, so it is clear that allocation scheme x is the decision variable of the model and the number of marks n is the number of decision variables.

2.2.2. Determination of the objective function. The expected objectives of the system are maximum utilization, minimum cost of material used, and minimum time spent on cutting process. Since the objectives contain both maximum and minimum values, they do not conform to the single optimization model of model (1). Therefore, here, considering the actual situation, all of them are transformed into a minimization problem, and the optimization model is transformed into $\min$, so that the number of objectives is $\rho = 3$.

One of the objectives of this paper is to decide which marks are used to cut the fabric and how many layers of each mark are allocated to satisfy the order requirements, while making the minimum cost of raw material and maximum utilization. Where the objective function for minimizing the cost of raw material is:

$$f_1(x) = \sum_{i=1}^{n} l_i \cdot x_i \cdot p_i \tag{2}$$

In order to convert the problem of maximum utilization of materials into a minimum value problem, here the problem of finding the maximum utilization is equivalently replaced by finding the minimum rate of loss, which can be obtained as the objective function:

$$f_2(x) = \frac{\sum_{i=1}^{n} (1-u_i) \cdot l_i \cdot x_i}{\sum_{i=1}^{n} l_i \cdot x_i} \tag{3}$$

The second objective of this paper is to minimize the time consumed in the cutting process, which includes the pulling time and the cutting time. In order to ensure the production conditions, $L_{\max} \geq 2L_{\min}$, the number of dividing beds c_i for the i nd mark is calculated as:

$$c_i = \begin{cases} \dfrac{x_i}{L_{\max}}, & \dfrac{x_i}{L_{\max}} \in \mathbb{Z} \\ \left\lceil \dfrac{x_i}{L_{\max}} \right\rceil, & \dfrac{x_i}{L_{\max}} \notin \mathbb{Z} \end{cases} \quad \left(\lceil * \rceil \text{indicates upward rounding}\right) \tag{4}$$

The overall shortest cropping time objective function is:

$$f_3(x) = \sum_{i=1}^{n} \left(c_i \cdot t_i + \frac{x_i \cdot l_i}{v_t} \right) \tag{5}$$

2.2.3. Determination of constraints. First, the allocation result should satisfy the customer's order quantity, i.e., the actual number of pieces kp_j required to be allocated for the j st size needs to satisfy equation (6):

$$d_j \leq kp_j \leq \left(d_j + \varphi_j\right)(j = 1, 2, \ldots, m) \tag{6}$$

where kp_j is determined by equation (7):

$$kp_j = \sum_{i=1}^{n} x_i \cdot a_{ij} \tag{7}$$

From this the constraint function can be constructed:

$$g_j(x) = \left(kp_j - d_j\right)\left(kp_j - d_j - \varphi_j\right)(j = 1, 2, \ldots, m) \tag{8}$$

Secondly, in order to facilitate the recycling of the cut pieces after cutting, the number of marks used should not be too many, and set its upper limit to the number of order sizes m, then:

$$count\{x_i \neq 0\} \leq m \tag{9}$$

From this, the constraint function is constructed:

$$g_j(x) = count\{x_i \neq 0\} - m(j = m+1) \tag{10}$$

Finally, the allocation result has to satisfy the bed-splitting production condition constraint, i.e., the number of Mark allocations used in any one sheet must not be less than the minimum number of cut layers $L_{\min}$, i.e:

$$x_i \geq L_{\min} \ or \ x_i = 0 \tag{11}$$

From the above analysis, the mathematical model of multi-objective optimization can be obtained as:

$$\min_{\{x_i\}} \begin{cases} f_1 = \sum_{i=1}^{n} l_i \cdot x_i \cdot p_i \\ f_2 = \dfrac{\sum_{i=1}^{n} (1-u_i) \cdot l_i \cdot x_i}{\sum_{i=1}^{n} l_i \cdot x_i} \\ f_3 = \sum_{i=1}^{n} \left(c(x_i) \cdot t_i + \dfrac{x_i \cdot l_i}{v_t} \right) \end{cases} \tag{12}$$

$$subjectto \begin{cases} g_j(x) = \left(kp_j - d_j\right)\left(kp_j - d_j - \varphi_j\right) \leq 0(j = 1, 2, \ldots, m) \\ g_j(x) = count\{x_i \neq 0\} - m \leq 0(j = m+1) \\ x_i \geq L_{\min} or x_i = 0 \\ x_i \in \square^{+} \end{cases} \tag{13}$$

2.3. APSO-based multi-objective optimization for solving CSM problems

In this section, a method for multi-objective optimal solution of CSM problem using Improved Fast Particle Swarm Algorithm (APSO) is proposed, which helps to improve the solution accuracy of the garment size matching problem and optimize the cutting process so as to reduce the cost of garment production.

2.3.1. Improved Fast Particle Swarm Algorithm (APSO):

1) Mathematical description. PSO algorithm is an efficient and intelligent optimization algorithm with fast search speed, high efficiency and simple algorithm [25]. It has the advantages of genetic algorithm (GA) and evolutionary planning to some extent. Compared with evolutionary planning algorithm, PSO algorithm is more stochastic. Compared to genetic algorithm (GA), it also undergoes constant

adjustment in the process of searching for the optimal value to be close to the individual optimal value and global optimal value.The basic principle of PSO is mainly to continuously complete the search for the optimal value of the individual extreme value and global extreme value, and the speed of retrieval is very fast in the process, however, as the number of iterations continues to increase, to a certain extent, it will make the particles fall into the situation of the local optimal value, which makes them always be in the the neighborhood of the local optimal solution, but cannot jump out of the loop.

In this paper, the improved fast particle swarm algorithm (APSO), which can adaptively adjust the inertia weights according to the adaptation value, is used to improve the performance of the algorithm. Since inertia weight ω is the most important parameter of the PSO algorithm in the process of optimization, the inertia weight ω is set reasonably in order to avoid the influence on the global and local optimization of the algorithm [26]. The value of inertia weight ω is usually taken in two cases: firstly, when its value is large, it has an absolute advantage in global optimization, and it is difficult to solve the global optimal solution because it is constantly expanding the target area of the search. The second is that when its value is small, it is advantageous for local optimization, in the case that the optimal solution is in the initial target search space, the PSO algorithm will find the global optimal solution in a short period of time, that is to say, all particles will be summarized quickly through iterative solving, and on the contrary, the inertia weights ω are very likely to be the local extreme value points. Therefore, the reasonable setting of inertia weights has absolute significance for the whole algorithm's optimization search process.

The inertia weights are updated according to the degree of early retrieval of the particles and the value of the adaptation degree is constantly adjusted to solve the problem of early retrieval of the algorithm and the easy oscillation near the global optimal solution at a later stage.

Let the fitness value of particle i be f_i and the optimal particle fitness value be f_m, then the average fitness value of the population is $f_{ave} = \frac{1}{n}\sum_{i=1}^{n} f_i$, which is defined as $\Delta = \left| f_i - f_{ave} \right|$. Then the inertia weights are updated as:

(1) When f_i is better than f_{ave}, then:

$$\omega = \omega_{\max} \tag{14}$$

(2) When f_i is superior to f_{ave} and inferior to f_m, the inertia weights remain unchanged.

(3) When f_i is inferior to f_{ave}, then:

$$\omega = \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min})(f - f_{\min})}{f_{ave} - f_{\min}} \tag{15}$$

where: $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum values of inertia weights, respectively, and f is the adaptation value of the current particle.

According to Eqs. (14) to (15), when the individual adaptation value of the particle is good, the particle whose adaptation value f_i is less than or equal to the average of the adaptation value of the current particle swarm is locally searched in the periphery of the optimal solution. When the individual fitness value of the particle is poor, that is, the fitness value f_i is greater than the average value of the current particle swarm fitness value of the particle, then by increasing the step length, in order to expand the target search area, in the shortest possible time to find the optimal solution with a variety of convergence is good, to prevent falling into the local optimal solution.

2) The basic flow of APSO algorithm. Compared with the basic particle swarm algorithm, APSO algorithm can not only inherit the simple and fast convergence characteristics of the basic particle swarm algorithm well, but also solves the trouble of easy to fall into the local optimal solution by

reasonably setting the inertia weights ω. The basic flowchart of the APSO algorithm is shown in Fig. 1, and its basic steps are as follows:

Step1: Initialize the parameters of the algorithm, the size of the population N, the particle acceleration factor c_1, c_2, the maximum and minimum inertia weights $\omega_{\max}$, $\omega_{\min}$, the maximum speed $V_{\max}$, the dimension of the particles m, etc..

Step2: Initialize the particle swarm position x_{ij} and velocity v_{ij} from the iteration condition, so the particle swarm position matrix is expressed as:

$$\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} & v_{11} & v_{12} & \cdots & v_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} & v_{21} & v_{22} & \cdots & v_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{N1} & x_{N2} & \cdots & x_{Nm} & v_{N1} & v_{N2} & \cdots & v_{Nm} \end{bmatrix} \tag{16}$$

where v_{ij} is a random number in the range of $\left[-V_{\max}, V_{\max}\right]$ intervals.

Step3: Calculate the fitness value of the particles.

Step4: Adjust the velocity and position of each particle in the population:

$$v_{ij} = \omega \cdot v_{ij} + c_1 \cdot r_1 \cdot \left(pb_{ij} - x_{ij}\right) + c_2 \cdot r_2 \cdot \left(gb_i - x_{ij}\right) \tag{17}$$

$$x_{ij} = x_{ij} + v_{ij} \tag{18}$$

where: r_1 and r_2 are $[0,1]$ uniformly distributed random numbers.

Step5: An inertia weight ω adaptive adjustment strategy is used to update the inertia weights, i.e., the inertia weights are adjusted according to the current adaptation values with the expression:

$$\omega = \begin{cases} \omega_{\min} + \dfrac{\left(\omega_{\max} - \omega_{\min}\right)\left(f - f_{\min}\right)}{f_{ave} - f_{\min}} & f \le f_{ave} \\ \omega_{\max} & f > f_{ave} \end{cases} \tag{19}$$

where: f is the adaptation value of the current particle; f_{ave} is the average value of the adaptation value; $f_{\min}$ is the minimum adaptation value.

Step6: Compare the current adaptation value f of each particle i with the historical optimal adaptation value p_{best}, if the current adaptation value f is less than p_{best}, update it to the historical optimal value, and thus find the global optimal value of the whole group.

Step7: Check the validity of the updated particles, and re-initialize them if they are in invalid state.

Step8: Determine whether the end conditions (one is the maximum number of iterations and the other is the deviation of the fitness value within the specified range) are reached. If yes, end the iteration, otherwise go to Step3 to continue the iterative solution.

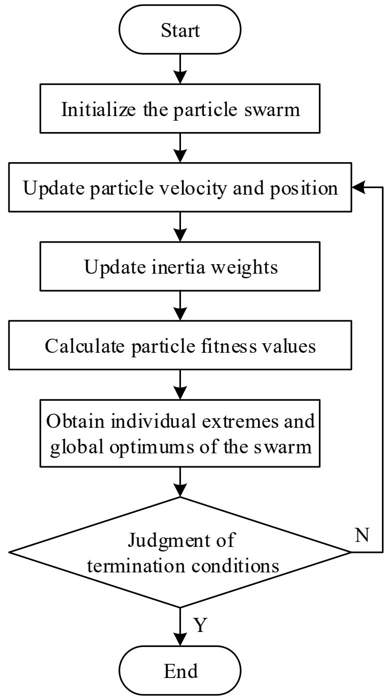

Fig. 1. Basic flow chart of APSO

2.3.2. Algorithm solving. The multi-objective optimization model of garment cutting size matching established in this paper belongs to the nonlinear constrained variable optimization problem, and the particle swarm algorithm has the characteristics of fast search speed, high efficiency, and simple algorithm, so it is widely used in the field of scientific research and design optimization. In this paper, the improved fast particle swarm algorithm (APSO), which can adaptively adjust the inertia weights, is used to improve the performance of the algorithm in order to search for the working condition transition point of the energy-saving speed curve that minimizes the cost of material and the cost of time, so as to achieve the optimization of cutting size matching of garments. The cutting size matching curve optimization model is used as a function of the fitness of the improved particle swarm algorithm, and the APSO algorithm is used to solve the different size matching condition transition points, compare the optimized production error and production cost, so as to obtain a more reasonable cutting size matching curve. The process of solving the optimization problem of garment cutting size matching is shown in Figure 2, and the specific steps are as follows:

1) Initialize the relevant parameters of the actual problem. By reasonably setting the relevant parameters, such as: particle swarm position (possible working condition transition points in garment cutting) and speed (the amount of change in production cost at each working condition transition point), etc. is the basis for solving the garment cutting size matching curve using the APSO algorithm.

2) Initialization of the algorithm population. Since the position of particles corresponds to the actual problem, the speed of particles is set randomly when initializing the population. Combined with the actual cutting size allocation of clothing orders, set the cutting size conditions of the allocation interval, and finally seek the optimization of the program with the lowest production cost in each sub-interval through the APSO algorithm.

3) Calculate the value of the fitness function. According to the actual data imported in (1) and the algorithm parameters in (2), the production cost C_i, the running time T_i and the target size S_i of

the corresponding subintervals of garment cutting size matching can be calculated according to the model.

When using particle swarm algorithm to solve the optimization problem with constraints, there is no uniform solution method, generally according to the actual problem and specific considerations, usually using the penalty function method or feasible solution transformation method to transform the constrained problem into an unconstrained problem. Based on the above constraints, this paper adopts the penalty function method, for the constraints of on-time and high-quality garment production, the fitness function can be expressed as:

$$f = C + f(T) + f(Q) \tag{20}$$

where C is the garment production costing model, $f(T)$ and $f(Q)$ represent the punctuality penalty function and the quality penalty function for garment cutting size matching respectively, and the following penalty rules are set:

If the cutting machine completes the garment CSM task within the specified time range, its function value is 0, i.e., in terms of punctuality penalty, it meets the requirements and no penalty is needed, on the contrary, if it exceeds the specified time, it will affect the customer's senses, while completing the garment production task before the specified time. Therefore, it can be seen from the analysis that there is a need to take certain penalties on the punctuality of the cutting machine to perform the CSM task, which can be expressed by the following equation (21):

$$f(T) = \alpha \left| \frac{T - T_D}{T_D} \right| \tag{21}$$

In the punctuality penalty function, the value of α takes the following equation:

$$\alpha = \begin{cases} 3 & T < T_D \\ 10 & T > T_D \end{cases} \tag{22}$$

The quality of clothing cutting size matching is commonly used to reflect the rate of change of the production error rate, according to the actual problem can be seen, the smaller the value of the production error rate, the higher the quality of clothing. However, in order to ensure that the garment cutting size matching production time under the premise of improving the quality of the garment, it is necessary to set a reasonable threshold for the production error rate, if greater than the threshold, that is, through the quality of the penalty function to penalize it, can be expressed in the following formula (23):

$$f(Q) = \beta \left| \frac{Q - a_{\lim}}{a_{\lim}} \right| \tag{23}$$

In the comfort penalty function, the value of β takes the following equation:

$$\beta = \begin{cases} 0 & Q \le a_{\lim} \\ 10 & Q > a_{\lim} \end{cases} \tag{24}$$

4) Updating inertia weights. In this paper, an inertia weight ω adaptive adjustment strategy is used, i.e., the inertia weights are adjusted according to the current adaptation values with the following expression:

$$\omega = \begin{cases} \omega_{\min} + \dfrac{(\omega_{\max} - \omega_{\min})(f - f_{\min})}{f_{ave} - f_{\min}} & f \le f_{ave} \\ \omega_{\max} & f > f_{ave} \end{cases} \tag{25}$$

where: f is the adaptation value of the current particle, f_{ave} is the average value of the adaptation value, and $f_{\min}$ is the minimum adaptation value.

5) Update strategy. Update the position of each particle in the group, use the current adaptation value p_{best} of each particle i to compare with the historical optimal adaptation value g_{best}, if the current adaptation value p_{best} is less than g_{best}, then update it to the historical optimal value, and thus find the global optimal value of the whole group.

6) Solving the global optimum. Through the adaptive weight-based APSO algorithm for the calculation of the objective function fitness value, and so on and so forth until it reaches the end conditions (one is the maximum number of iterations, the other is the deviation of the fitness value within the specified range), and finally get the production cost sequence that satisfies the constraints.

7) Generate the cost curve of optimal garment cutting size matching. Based on the above operation process realizes the optimization search for the optimal garment cutting size matching scheme of the cutting machine operation in each sub-interval, and then obtains the cost curve of cutting size matching of the cutting machine in the process of garment generation.

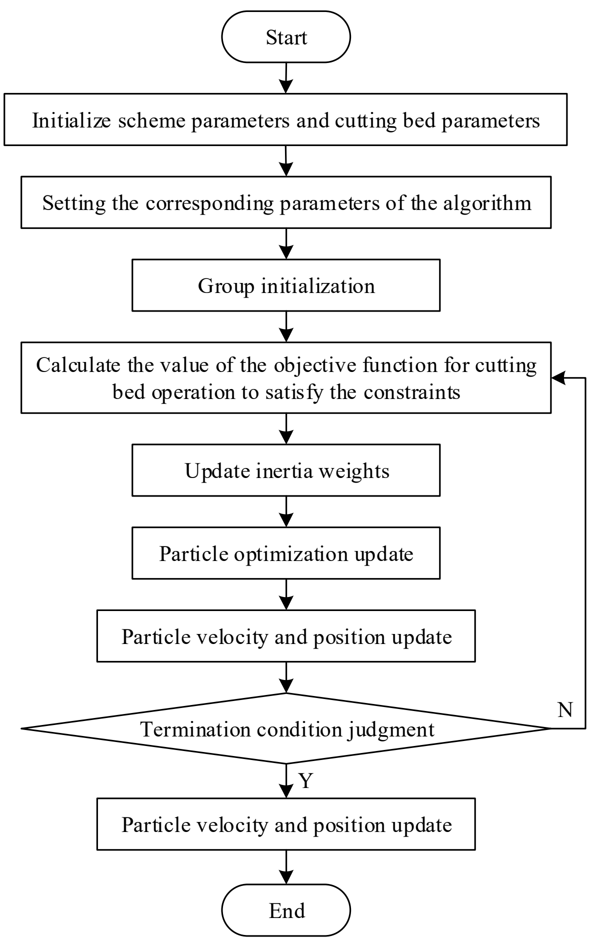

Fig. 2. APSO algorithm solution process

3. Model application analysis

3.1. Algorithm experiment and its result analysis

In order to verify the effectiveness of the APSO algorithm, this paper takes the standard particle swarm algorithm (PSO), inertia weight linearly decreasing particle swarm algorithm (LDWPSO) [27] as a comparison, and the main parameters of the algorithm are set as follows:

Particle population size $N=40$, Maximum number of iterations $Iter_{\max}=1200$, Learning factor $C1=C_2=2$, In PSO algorithm $w=0.8$, In LDWPSO algorithm $w_{\max}=0.9$, $w_{\min}=0.4$, The hardware and software environments used for the experiments were Windows 11 Professional 22H2, Intel(R) Core(TM) i7-8750H CPU@2.20GHz, memory 16.0 GB DDR4 2667Mhz, tested in Matlab R2022 language environment.

3.1.1. Griewank function tests. The Griewank function is a multimodal test function composed of quadratically convex and oscillating nonconvex functions, and its optimization search difficulty first becomes harder and then easier as the function dimension increases. In order to more effectively test the performance of the APSO algorithm, the test APSO algorithm and the comparison algorithm in the problem dimension from 10 to 30, each algorithm independently run 30 times to obtain the optimal fitness value average, minimum, maximum and standard deviation data shown in Table 1. And when dimension $D=30$, the curves of the three algorithms' optimal fitness values with the number of iterations in a certain independent run are shown in Fig. 3. For easy observation, the vertical coordinate in the figure is the logarithm of the optimal fitness value taken as the base 10.

From the data in Table 1, it can be seen that in problem dimension $D=10$, all three algorithms can achieve better optimization results, the optimal solution is found by the APSO algorithm, and the optimal fitness values of the APSO algorithm in the 30 times of optimization are better than the average value, the maximum value, and the standard deviation of the PSO and the LDWPSO.With the increase of the problem dimensions, the optimization accuracy of the PSO and the LDWPSO decreases significantly, while the APSO algorithm still maintains a good optimization accuracy.

In problem dimension $D=10,20,30$, the APSO algorithm finds the global optimal solution 13 times, 8 times, and 5 times during 30 independent runs, while the other algorithms fail to find the global optimum. It can be seen that on the test on Griewank function, the APSO algorithm has a significant improvement in the accuracy and stability of finding the optimum compared to PSO and LDWPSO algorithms.

Table 1. Test comparison results on the Griewank function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	0.1759	0.0661	0.3896	0.0882
	LDWPSO	0.1747	0.0574	0.2938	0.0716
	APSO	0.0152	0.0017	0.0578	0.0208
D=20	PSO	0.5853	0.1642	1.0567	0.3013
	LDWPSO	0.1645	0.0755	0.2748	0.0601
	APSO	0.0152	0.0017	0.0578	0.0208
D=30	PSO	0.9788	0.8237	1.4099	0.1955
	LDWPSO	0.6807	0.5467	0.9208	0.1011
	APSO	0.0011	4.01E-4	0.0592	0.0152

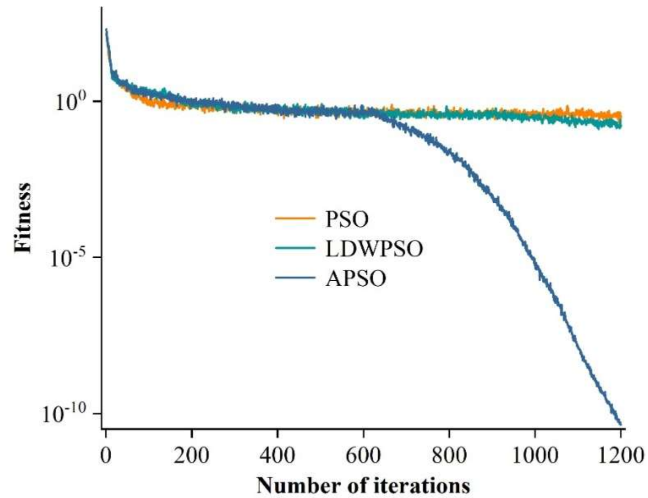

Fig. 3. Curve of optimal fitness value with the number of iterations

3.1.2. Ackle function test. Ackle function is a continuous function consisting of an exponential function superimposed on a moderately enlarged cosine function, this function search when the algorithm is easy to fall into the trap of the local optimum, only the algorithms with strong global search capabilities can gradually find the optimum point. In accordance with the same method using PSO, LDWPSO, APSO for testing, Ackle function test results are shown in Table 2.

From the data in Table 2, it can be seen that as the problem dimension increases, the optimization accuracy of the three algorithms decreases, PSO and LDWPSO in the case of the same problem dimension of the algorithm performance difference is small, while the performance of the APSO algorithm to be more significant.

Table 2. The test results of three algorithms on Ackle function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	0.8107	0.1064	1.6652	0.6633
	LDWPSO	0.7429	0.0015	2.0178	0.8378
	APSO	2.02E-15	2.02E-15	2.02E-15	0
D=20	PSO	2.2640	1.6604	3.1005	0.3864
	LDWPSO	2.9892	2.1799	3.4932	0.3801
	APSO	2.08E-10	3.17E-11	8.01E-10	1.97E-10
D=30	PSO	3.9028	2.7914	4.8385	0.7004
	LDWPSO	3.0693	2.6267	4.0569	0.4398
	APSO	4.01E-7	2.39E-7	8.12E-7	1.40E-7

The fitness variation curves of the three algorithms in Ackle function for problem dimension $D=30$ are shown in Fig. 4. From Fig. 4, it can be seen that when PSO and LDWPSO algorithms fall into the local optimal solution, APSO still maintains a better exploration ability.

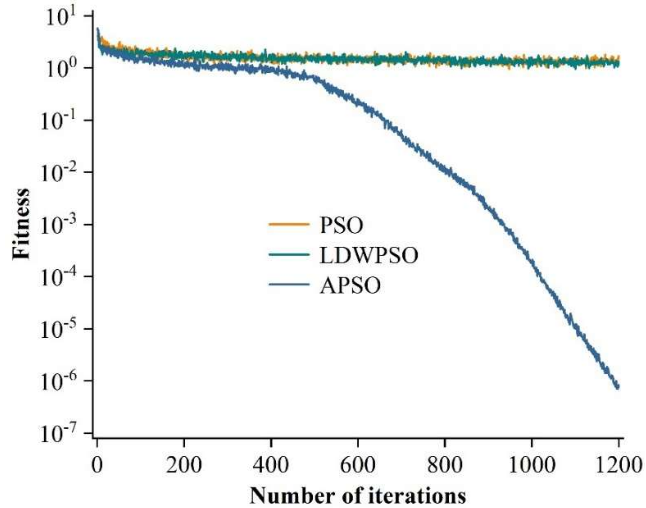

Fig. 4. Fitness curve of three algorithms on Ackle function 30 dimension

3.1.3. Levy function test. The Levy function is an asymmetric multimode test function with multiple oscillating local extreme points. The Levy function test results of the three algorithms are shown in Table 3.

From Table 3, it can be seen that PSO and LDWPSO lack the ability to jump out of the local extremes, and are easy to fall into the local optimal solution during the optimization process, and the optimization accuracy decreases as the problem dimension increases.

Table 3. The test results of three algorithms on Levy function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	0.1267	1.78E-11	0.9979	0.2813
	LDWPSO	0.4852	5.64E-12	3.4707	0.9410
	APSO	1.48E-32	1.62E-32	1.49E-32	0
D=20	PSO	2.2675	1.6596	3.0988	0.3877
	LDWPSO	2.9936	2.1826	3.4947	0.3813
	APSO	8.06E-20	7.26E-21	6.42E-19	1.69E-19
D=30	PSO	2.8784	0.5495	6.1237	1.7023
	LDWPSO	1.4092	0.0002	4.8873	1.3256
	APSO	5.48E-13	3.07E-14	4.57E-12	9.72E-13

The adaptive change curves of the three algorithms on the 30 dimensions of the Levy function are shown in Fig. 5. From Fig. 5, it can be seen that APSO maintains a certain global search ability while accelerating the convergence speed by adaptive change of inertia weights, so as to better find the optimal solution.

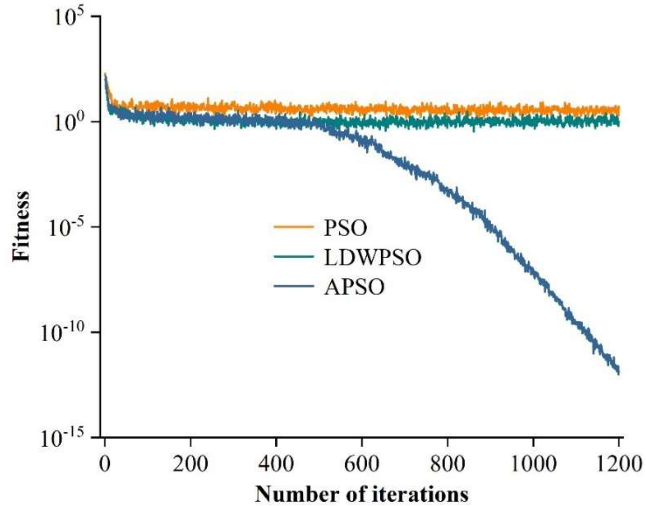

Fig. 5. Fitness curve of three algorithms on Levy function 30 dimension

3.1.4. Rastrign function test. The Rastrign function is characterized by the fact that the positions of the minima occur regularly and can be used to check the validity of the arithmetic placement method when there is some regularity in the solution. The results of the Rastrign function test for the three algorithms are shown in Table 4 as well as Figure 6. It can be seen that the performance of APSO in all dimensions are more excellent, in the lower dimension can find the global optimum, in the dimension after the increase in the solution accuracy only a little decline, Figure 6 can be seen that the APSO algorithm compared to the two algorithms at the beginning of the convergence of faster, and its accuracy advantage is also very obvious.

Table 4. The test results of three algorithms on Rastrign function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	5.1482	2.9483	7.2818	1.0252
	LDWPSO	6.7316	2.1995	9.9751	2.1787
	APSO	0	0	0	0
D=20	PSO	25.1962	15.0377	39.6928	7.3892
	LDWPSO	27.5336	18.6679	40.91	7.6225
	APSO	7.24E-21	2.56E-23	1.98E-20	5.37E-21
D=30	PSO	52.3801	37.7856	72.0576	9.5182
	LDWPSO	39.1177	22.8414	54.0139	7.6933
	APSO	1.35E-13	4.89E-15	6.86E-13	1.73E-13

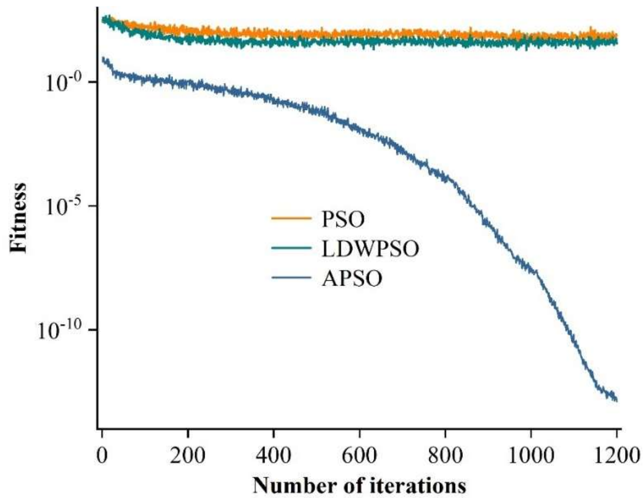

Fig. 6. Fitness curve of three algorithms on Rastrign function 30 dimension

3.1.5. Schwefel function test. Schwefel is a complex multi-peak function that contains a large number of local extreme value regions, which makes it difficult to find the optimal solution, and the optimal solution is located at $(504.879,\ldots,504.879)$. The three algorithms optimize the Schwefel function to obtain the test results as shown in Table 5 and Fig. 7. From the data in the table, it can be seen that the performance advantage of APSO over PSO and LDWPSO in the Schwefel function test is still very obvious, but the three algorithms fail to solve the global optimum.

Table 5. The test results of three algorithms on Schwefel function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	1832.0886	1236.3436	2500.7764	336.6050
	LDWPSO	1785.9159	1018.1676	2342.4951	341.4989
	APSO	517.0668	245.5274	941.8646	193.4507
D=20	PSO	4038.2114	3287.8675	4817.8412	451.4027
	LDWPSO	3709.7307	2226.0814	4809.7845	696.7202
	APSO	1485.9912	879.6022	2066.5765	328.9955
D=30	PSO	6168.6214	3544.5741	7036.3187	871.0483
	LDWPSO	5656.3999	4688.8431	6956.0645	747.9959
	APSO	3013.4843	218.0857	4165.3657	437.9539

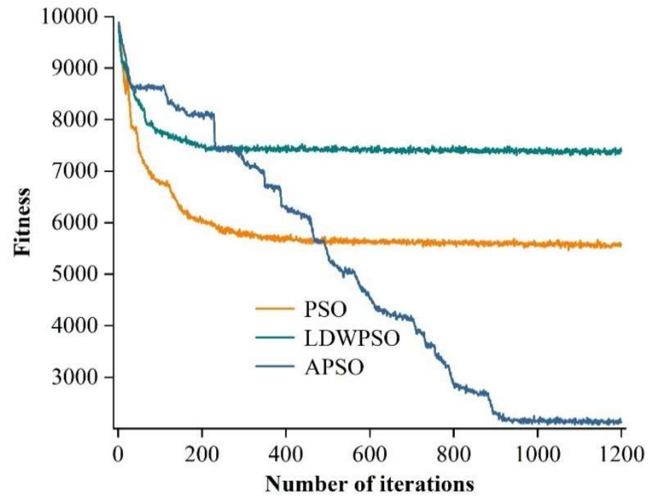

Fig. 7. Fitness curve of three algorithms on Schwefel function 30 dimension

By observing the positions of the globally optimal particles in the APSO algorithm in the test when the dimension is equal to 10 as shown in Table 6, it can be seen that the optimal particles of the APSO algorithm reach the optimal positions in all 7 dimensions, while PSO and LDWPSO reach the optimal positions in 1 dimension and 2 dimensions, respectively.

Table 6. Global optimal particle position of APSO on the Schwefel function in dimension 10

	1	2	3	4	5	6	7	8	9	10
1	504.879	-600	504.8796	504.8796	-373.614	504.8796	504.8796	-600	504.879	504.879
2	504.879	-600	504.879	504.879	-373.614	504.8796	504.8796	-600	504.879	504.879
3	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
4	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
5	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
6	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
7	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
8	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
9	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
10	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
11	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
12	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
13	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
14	504.879	-600	504.879	504.879	-373.614	504.879	504.879	-600	504.879	504.879
15	504.8796	-600	504.8796	504.8796	-373.614	504.8796	504.8796	-600	504.8796	504.8796
16	504.8796	-600	504.8796	504.8796	-373.614	504.8796	504.8796	-600	504.8796	504.8796

3.1.6. Sphere function tests. The Sphere function is a single-peaked spherical function with only one global minimum, and the use of Sphere enables better testing of the algorithms' optimization accuracy. The results of testing the three algorithms using the Sphere function are shown in Table 7. According to the data in the table, it can be seen that the optimization accuracy of APSO in solving the Sphere function has an obvious advantage over PSO and LDWPSO, because the Sphere function has only one global minimum, the optimization accuracy of the particle swarm algorithm with linearly decreasing inertia weights decreases compared to the particle swarm algorithm with fixed inertia weights, whereas the optimization accuracy of the APSO algorithm decreases by using the self-adaptive inertia weights and the nonlinear asynchronous The optimization accuracy of APSO algorithm is effectively improved by the adaptive inertia weights and nonlinear asynchronous learning factor strategy.

Table 7. The test results of three algorithms on Sphere function

Problem dimension	Algorithm	Mean value	Minimum value	Maximum value	Standard deviation
D=10	PSO	7.14E-11	6.07E-12	6.49E-10	1.72E-10
	LDWPSO	1.34E-9	2.31E-15	1.15E-8	2.94E-9
	APSO	2.57E-37	1.71E-40	6.44E-37	1.96E-37
D=20	PSO	8.36E-5	1.24E-5	2.76E-4	7.21E-5
	LDWPSO	2.74E-3	1.42E-3	3.35E-3	9.82E-4
	APSO	1.61E-20	2.94E-37	1.52E-19	3.41E+2
D=30	PSO	2.52E-3	1.21E-3	4.46E-3	1.25E-3
	LDWPSO	2.18E-2	6.24E-3	6.77E-3	1.63E-2
	APSO	1.17E-13	1.91E-15	9.75E-13	2.54E-13

The adaptation curves of the three algorithms in the Sphere function at problem dimension $D=30$ are shown in Fig. 8. From the figure, it can be seen that the APSO algorithm converges slower in the early stage compared to the PSO and LDWPSO algorithms, but with the increase in the number of iterations, the APSO algorithm's in the convergence speed, the more obvious the accuracy of the search for excellence.

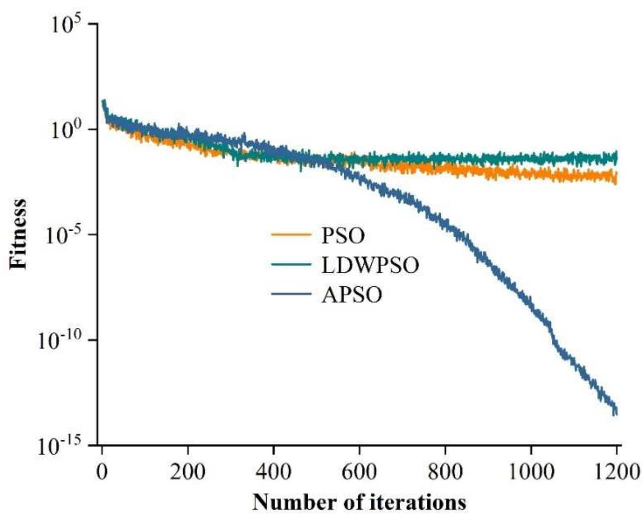

Fig. 8. Fitness curve of three algorithms on Sphere function 30 dimension

3.2. *Application Cases and Comparative Analysis*

In this section, in response to an enterprise's need for an implementation plan for the customized multicolor clothing order cut-to-size matching (CSM) problem, the proposed APSO optimization algorithm is used to optimize the solution of the given four different clothing orders, and a comparative analysis is conducted with the LINGO optimization software, which has been applied more often in the existing research, as well as with other particle swarm algorithms. Comparative analysis is carried out. Two parts of the experiment are given: (1) APSO algorithm applied to 4 different orders. (2) Comparative experiments of APSO, SVM-PSO, PSO, GA-PSO, LINGO, LDWPSO and other algorithms.

The experimental parameters are given as follows: the maximum number of available cutting machines $M_{\max}=10$, the maximum number of layers $L_{\max}=140$, the maximum number of marks allowed to be placed on a single cutting machine $B_{\max}=10$, the maximum number of pieces allowed to be placed on a single size $B^{s}_{\max} = 3$, the maximum production error allowed by the enterprise $E_a = 3\%$, and the number of iterations $Iter = 6000$. During the experiment, the iteration is terminated when the production error is satisfied by the enterprise for $t_n = 30$ repetitions.

3.2.1. APSO Algorithm Application Cases. The color, size specification and quantity demanded for each clothing order are shown in Table 8. Among them, Order 1 is a regular order, Order 2 has the same quantity of each size under three colors, and Order 3 and Order 4 are irregular orders, the latter with 1 more color.

Table 8. 4 clothing orders

Order number	Color	Size					
		XS	S	M	L	XL	XXL
1	White	110	210	210	210	110	0
	Red	160	260	240	220	90	0
	Blue	110	240	240	150	130	0
2	White	60	132	154	221	108	56
	Red	60	132	154	221	108	56
	Blue	60	132	154	221	108	56
3	Black	91	246	150	143	217	0
	Cream-colored	83	119	242	225	157	0
	Yellow	74	101	132	229	118	0
	Blue	71	123	158	275	138	0
4	Black	112	171	227	255	82	0
	Cream-colored	116	193	226	247	37	0
	Yellow	69	128	137	74	26	0
	Blue	122	88	228	161	84	0
	Gray	92	148	253	152	113	0

APSO is used to solve the above CSM problem for different orders with different specifications, and the results of the first 5 sets of solutions obtained from the optimal solution set of the algorithm at a certain time are shown in Table 9.

For order 1, the manual empirical method is used for cutting and dividing the beds, and each color is cut separately, and 7 cutting machines are needed to achieve the difference-free production when the production constraints are met, corresponding to the size combination scheme B_{O1} and the number of layers of fabric scheme L_{O1}, respectively:

$$B_{O1} = \begin{bmatrix} 1 & 1 & 3 & 2 & 1 \\ 1 & 2 & 3 & 3 & 1 \\ 8 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 4 & 6 \\ 1 & 3 & 1 & 2 & 2 \\ 0 & 4 & 3 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}, L_{O1} = \begin{bmatrix} 120 & 0 & 0 \\ 0 & 90 & 0 \\ 0 & 10 & 0 \\ 0 & 20 & 0 \\ 0 & 0 & 120 \\ 0 & 0 & 10 \\ 0 & 0 & 10 \end{bmatrix} \tag{26}$$

The optimal result obtained by the APSO algorithm requires only 4 cutting machines (as shown in Scheme 1 of Order 1 in Table 9), and the corresponding size combinations Scheme B_{O1} and the number of fabric layers Scheme L_{O1} are:

$$B_{O1}=\begin{bmatrix}1&3&4&2\\1&3&3&2\\3&3&3&1\\1&4&3&2\end{bmatrix},L_{O1}=\begin{bmatrix}0&3&47\\60&42&38\\0&45&5\\60&30&30\end{bmatrix} \tag{27}$$

According to the comparison of the above results, the APSO algorithm makes a more reasonable allocation of size combinations and the number of fabric layers, thus reducing the number of cutting beds used. As shown in Table 9, the APSO algorithm obtains superior optimal CSM solutions for four different types of apparel orders, and the cutting production errors are much smaller than the enterprise requirement indexes, especially for large-scale customized irregular orders, the scored bed solution is very reasonable, and such orders rely on manual experience to solve very difficult, and the error is often very large. Obviously, the larger the scale and irregularity of the order, the production error rate is relatively more difficult to control. In addition, among the five cutting and dividing solutions for order 4, solution 1 has a smaller production error, and solution 3 uses a smaller number of cutting machines, so the manager of the cutting department can weigh the two cutting solutions according to the cost of fabric and customer requirements, and assume that the cost of fabric is lower, then solution 3 can be chosen as the final cutting and dividing solution, which will make the overall cost optimal.

Table 9. The first 5 group solutions for the optimal set of 4 clothing orders

Order number	Number of clothes	Scheme	Error /piece	Number of cutting beds	Error rate /%
1	2690	1	0	4	0
		2	0	6	0
		3	4	5	0.15
		4	4	6	0.15
		5	4	7	0.15
2	2193	1	0	5	0
		2	0	9	0
		3	4	10	0.18
		4	8	5	0.36
		5	10	6	0.46
3	3092	1	6	9	0.19
		2	7	7	0.23
		3	10	5	0.32
		4	12	8	0.39
		5	12	10	0.39
4	3541	1	7	8	0.20
		2	14	9	0.40
		3	15	7	0.42
		4	15	9	0.42
		5	18	10	0.51

3.2.2. Comparative analysis. In order to compare the superiority of the proposed APSO algorithm, it is compared and analyzed with SVM-PSO [28], PSO, GA-PSO [29], LINGO, and LDWPSO algorithms, which are more widely used in current research.

Taking garment order 4 as an example, APSO, SVM-PSO, PSO, GA-PSO, LINGO, and LDWPSO are respectively used to solve its CSM problem, and each algorithm is run for 60 times, and five of them are recorded as the relative optimal cutting and bed-splitting schemes, and the experimental results are obtained as shown in Table 10.

Table 10. Algorithm comparison experiment results for order 4

Optimization algorithm	Scheme	Error /piece	Number of cutting beds	Error rate /%	Running time/s	Number of iterations
PSO	1	69	10	1.95	7.68	1238
	2	71	9	2.01	11.45	1741
	3	60	8	1.69	19.6	2775
	4	91	7	2.57	35.54	6000
	5	160	6	4.52	35.67	6000
SVM-PSO	1	17	9	0.48	9.24	30
	2	16	7	0.45	16.95	30
	3	15	7	0.42	10.92	30
	4	21	8	0.59	10.34	30
	5	19	6	0.54	10.07	30
LDWPSO	1	101	10	2.85	65.02	6000
	2	169	9	4.77	68.69	6000
	3	197	8	5.56	65.12	6000
	4	244	7	6.89	66.34	6000
	5	284	6	8.02	68.07	6000
GA-PSO	1	45	10	1.27	7.64	225
	2	59	9	1.67	9.78	273
	3	99	8	2.80	83.79	6000
	4	106	7	2.99	90.29	6000
	5	169	6	4.77	90.61	6000
APSO	1	6	9	0.17	3.48	30
	2	8	8	0.23	3.62	30
	3	7	7	0.20	3.86	30
	4	7	8	0.20	3.63	30
	5	10	9	0.28	3.66	30
LINGO	1	21	9	0.59	-	45
	2	9	8	0.25	-	73
	3	99	7	2.80	-	105
	4	95	8	2.68	-	61
	5	91	9	2.57	-	181

In order to compare the optimization performance of APSO algorithm more clearly, the Pareto optimal solution sets of APSO and GA-PSO algorithms are given as shown in Fig. 9. Since the PSO algorithm will converge to the optimal solution set with the same number of cutting beds, the optimization results of GA-PSO algorithm under different number of cutting beds in Table 10 are directly plotted in the figure, and the Pareto optimal front-end curves of Scheme 1 under the APSO algorithm are also given.

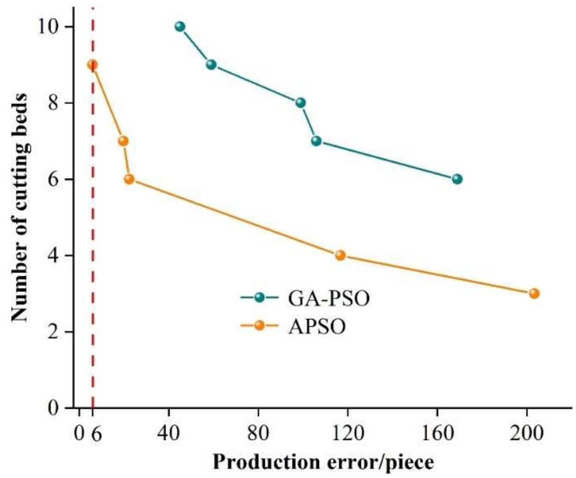

Fig. 9. The Pareto optimal solution set for APSO versus GA-PSO algorithm

From the results in Fig. 9 and Table 10, it can be seen that the PSO algorithm obtains a feasible solution that meets the requirements of the enterprise when the number of cutting beds is 8 to 10, and the GA-PSO obtains a feasible solution when the number of cutting beds is 9 and 10, and the LDWPSO algorithm does not obtain a feasible solution within the specified range of the number of cutting beds, and the optimization results have not been improved by conducting several sets of tests on its parameters within a reasonable range during the experiment, so it can be seen that LDWPSO algorithm has limited optimization performance in solving complex CSM problems. On the other hand, the SVM-PSO algorithm and the APSO algorithm can be solved to obtain a reasonable CSM scheme every time, and the production error in the five groups of schemes obtained by the latter is no more than 0.30%, which is much smaller than the error requirements set by enterprises. Meanwhile, APSO algorithm is significantly better than SVM-PSO algorithm in terms of production error and solution efficiency, and the minimum production error of the former is 6 pieces, and the minimum time for obtaining feasible solutions is 3.48 s. From the experimental results, it can be concluded that APSO algorithm can effectively simplify the algorithm coding and the SCM solving process, and improve the algorithm solving accuracy and algorithm efficiency. In addition, in the LINGO optimization software, the number of cutting beds and size combinations in the five groups of solutions obtained by the APSO algorithm are given as known information, and the corresponding number of layers and production results are solved, and only two results meet the production requirements of the enterprise, and the error rates are larger.

In summary, the proposed APSO algorithm has better optimization accuracy and solving efficiency than other improved particle swarm algorithms and optimization software in optimizing and solving the problem of matching the size of large-volume irregular apparel orders, and it can obtain a more superior cutting and dividing scheme.

4. Conclusion

In this paper, a multi-objective optimization model is constructed for the clothing cutting size matching problem, and the improved fast particle swarm algorithm (APSO) is used to realize the optimal solution of the model.

The test results of APSO algorithm and PSO algorithm and LDWPSO algorithm in six functions, Griewank, Ackle, Levy, Rastrign, Schwefel and Sphere, show that the optimization accuracy of the three algorithms decreases with the increase of the problem dimensions, but the decrease of APSO algorithm is much smaller than that of the PSO algorithm and the LDWPSO algorithms, and is able to maintain better exploration ability when PSO and LDWPSO algorithms fall into local optimal solutions.

In the case application process, the APSO algorithm is able to make a more reasonable allocation of the size combination and the number of layers of fabric, reduce the number of cutting beds used, and optimize the overall cost. For four different types of garment orders, APSO algorithm obtains the optimal solution set of superior garment cutting size allocation scheme, and the cutting production error is far less than 3% of the enterprise requirement index.

Compared with PSO algorithm, GA-PSO algorithm, LDWPSO algorithm, SVM-PSO algorithm and APSO algorithm can be solved every time to get a more reasonable optimization scheme, and the production error of the five groups of solutions obtained by APSO algorithm is less than 0.30%, which is far less than the error requirements set by the enterprise. At the same time, the APSO algorithm is significantly better than the SVM-PSO algorithm in terms of production error and solution efficiency, the minimum production error is only 6 pieces, and the minimum time for obtaining a feasible solution is 3.48 s. In addition, there are only two results in the LINGO optimization software to meet the production requirements of the enterprise, and the error rates are larger. It can be seen that the APSO algorithm proposed in this paper has better optimization accuracy and solving efficiency in the optimization solution for the large-volume irregular garment order size cutting matching problem, and can obtain a more superior cutting and bed splitting scheme.

References

[1] Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M., & Gandomi, A. H. (2021). The arithmetic optimization algorithm. *Computer methods in applied mechanics and engineering*, 376, 113609.

[2] Beiranvand, V., Hare, W., & Lucet, Y. (2017). Best practices for comparing optimization algorithms. *Optimization and Engineering*, 18, 815-848.

[3] Deng, W., Zhao, H., Zou, L., Li, G., Yang, X., & Wu, D. (2017). A novel collaborative optimization algorithm in solving complex optimization problems. *Soft Computing*, 21, 4387-4398.

[4] Gad, A. G. (2022). Particle swarm optimization algorithm and its applications: a systematic review. *Archives of computational methods in engineering*, 29(5), 2531-2561.

[5] Ab Wahab, M. N., Nefti-Meziani, S., & Atyabi, A. (2015). A comprehensive review of swarm optimization algorithms. *PloS one*, 10(5), e0122827.

[6] Jia, H., Peng, X., & Lang, C. (2021). Remora optimization algorithm. *Expert Systems with Applications*, 185, 115665.

[7] Gharehchopogh, F. S., & Gholizadeh, H. (2019). A comprehensive survey: Whale Optimization Algorithm and its applications. *Swarm and Evolutionary Computation*, 48, 1-24.

[8] Kaveh, A., & Dadras, A. (2017). A novel meta-heuristic optimization algorithm: thermal exchange optimization. *Advances in engineering software*, 110, 69-84.

[9] Lee, S., & Kwon, S. (2020). A study on the dress form for the making dresses: Focusing on the size cover rate and correction. *Human Ecology Research*, 58(2), 215-228.

[10] Hidayati, S. C., Hsu, C. C., Chang, Y. T., Hua, K. L., Fu, J., & Cheng, W. H. (2018, October). What dress fits me best? Fashion recommendation on the clothing style for personal body shape. *In Proceedings of the 26th ACM international conference on Multimedia* (pp. 438-446).

[11] Davidson, A., & Gundlach, E. (2023). Creating predictive clothing size models for online customers. *International Journal of Mathematical Education in Science and Technology*, 54(4), 614-629.

[12] Mahoney, C. (2022). The Effects of Clothing Size on Self-Esteem regarding Gender. *Perspectives*, 14(1), 2.

[13] Bishop, K., Gruys, K., & Evans, M. (2018). Sized out: Women, clothing size, and inequality. *Gender & Society*, 32(2), 180-203.

[14] Tiwari, G., Bhatnagar, B. L., Tung, T., & Pons-Moll, G. (2020). Sizer: A dataset and model for parsing 3d clothing and learning size sensitive 3d clothing. *In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16* (pp. 1-18). Springer International Publishing.

[15] An, D. J., & Lee, J. Y. (2018). Survey on the brand of online custom dress shirts and analysis of the sizing system. *Fashion & Textile Research Journal*, 20(5), 556-568.

[16] Romans, M. K. (2018). Dress Rules and Body Perceptions of Mature Plus Size Women. *Louisiana State University and Agricultural & Mechanical College*.

[17] Nasibov, E., Demir, M., & Vahaplar, A. (2019). A fuzzy logic apparel size decision methodology for online marketing. *International Journal of Clothing Science and Technology*, 31(2), 299-315.

[18] Cundall, A., & Guo, K. (2017). Women gaze behaviour in assessing female bodies: the effects of clothing, body size, own body composition and body satisfaction. *Psychological Research*, 81, 1-12.

[19] Liu, F. (2022). The importance of garment specification on formal dress: The application of a perfect wedding dress. *Art And Society*, 1(2), 41-55.

[20] Lee, S., & Kwon, S. (2023). Dress Design Analysis and Pattern Making Techniques for Increasing the Efficiency of Dress Production. *Human Ecology Research*, 61(1), 53-71.

[21] Yang, X., He, X., Wang, X., Ma, Y., Feng, F., Wang, M., & Chua, T. S. (2019, July). Interpretable fashion matching with rich attributes. *In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 775-784).

[22] Witzel, C., O'Regan, J. K., & Hansmann-Roth, S. (2017). The dress and individual differences in the perception of surface properties. *Vision Research*, 141, 76-94.

[23] Carufel, R., & Bye, E. (2020). Exploration of the body–garment relationship theory through the analysis of a sheath dress. *Fashion and Textiles*, 7, 1-28.

[24] Dabolina, I., Silina, L., & Apse-Apsitis, P. (2018, December). Evaluation of clothing fit. *In IOP Conference Series: Materials Science and Engineering* (Vol. 459, No. 1, p. 012077). IOP Publishing.

[25] Urbanus Mwanzia Ngei, Abraham Mutunga Nyete, Peter Musau Moses & Cyrus Wekesa. (2024). Optimal sizing and placement of STATCOM, TCSC and UPFC using a novel hybrid genetic algorithm-improved particle swarm optimization. Heliyon(23),e40682-e40682.

[26] Mojtaba Rezaei, Seyed Mohammad Hassan Hosseini & Bizhan Nemati. (2024). Presenting an optimal energy management model in microgrids using APSO algorithm. Electrical Engineering(4),4665-4677.

[27] Liu Fuzheng, Gao Junwei & Liu Huabo. (2020). The Feature Extraction and Diagnosis of Rolling Bearing Based on CEEMD and LDWPSO-PNN.IEEE Access19810-19819.

[28] Lama K. Alsaykhan & Mashael S. Maashi. (2024). A hybrid detection model for acute lymphocytic leukemia using support vector machine and particle swarm optimization (SVM-PSO).Scientific Reports(1),23483-23483.

[29] Muhammad Saad, Rabia Noor Enam & Rehan Qureshi. (2024). Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application. Frontiers in Big Data1358486-1358486.