Traffic & Transportation

An Improved K-means Clustering Algorithm Based on EIQ Analysis for Order Batching of Shuttle-Based Storage/Retrieval Systems

Chuanjun CHEN[1], Hongqiang FAN[2], Junjie LIU[3], Shun LI[4]

Original Scientific Paper
Submitted: 28 April 2025
Accepted: 27 July 2025
Published: 28 Apr 2026

[1] Corresponding author, chencj20@mails.tsinghua.edu.cn, Department of Automation, Tsinghua University, Beijing, China
[2] fanhongqiang@bupt.edu.cn, Department of Intelligent Engineering and Automation, Beijing University of Posts and Telecommunications, Beijing, China
[3] liusheng0@yeah.net, BZS (Beijing) Technology Development Co., Ltd., Beijing, China
[4] lishun@bzkj.cn, BZS (Beijing) Technology Development Co., Ltd., Beijing, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT
Shuttle-based storage/retrieval systems (SBS/RS) require efficient order batching to optimise split-case picking. Original K-means clustering, which groups orders based on overlapping SKUs to minimise bin presentations, struggles with high-dimensional, sparse pharmaceutical data due to computational inefficiency, unsuitable distance metrics and unstable initialisation. We propose an enhanced K-means algorithm based on EIQ analysis. High-frequency SKUs are selected using IK frequency filtering, while Pearson correlation is applied to remove redundant features and reduce dimensionality. Cluster centre initialisation is improved using a roulette-based strategy, and cosine distance replaces Euclidean distance to better capture SKU similarity. Case studies using real data from Company A show that the proposed method outperforms both first-come-first-serve (FCFS) and standard K-means in reducing bin presentations and enhancing processing stability. The algorithm remains robust regardless of SKU popularity shifts. Sensitivity analysis confirms strong performance within appropriate thresholds for feature selection (n: 20–25) and correlation filtering (Pearson correlation: 0.8–0.9). Furthermore, as the number of item-lines per order increases, the improved algorithm yields greater efficiency gains. This algorithm can also be well applied to other industries.

KEYWORDS
order batching; SBS/RS; pharmaceutical order; EIQ; improved K-means clustering.

1. INTRODUCTION

In China, with the deepening of pharmaceutical policy reform, including the hierarchical medical system, zero-markup medicine policy and national centralised medicine purchase, pharmaceutical circulation enterprises are facing a change in business structure, marked by an increase in split-case picking orders and a decrease in case orders. The traditional person-to-goods picking mode has been difficult to adapt to the new demand due to a high error rate and high labour costs, which promoted the rapid development of the goods-to-person picking system. As a representative of the goods-to-person mode, the SBS/RS offers high-density storage and efficient retrieval. The SBS/RS consists of multi-layer storage shelves, shuttles, elevators, conveyors and control systems. The bin presentation is completed by the elevators and the shuttle, and the order picking operation is carried out on the picking stations. However, the space of the picking stations is limited, and it is impossible to configure individual boxes for each order. Therefore, it is necessary to realise the importance of order batching. If centralised order processing causes containers to be dispatched in bulk from the SBS/RS, then the conveyor belt will become congested due to the excessive number of SKU bins required by orders external to the picking station. In contrast, the decentralised processing will reduce the equipment utilisation because SKU bins are spatially dispersed within the SBS/RS. Therefore, developing an

efficient order batching algorithm has become a critical step in enhancing the performance of the SBS/RS. A well-designed order batching strategy can significantly improve system efficiency by maximising overlaps in SKU requirements and minimising redundant bin retrievals.

The research on the order batching problem in warehouse management can be traced back to 1973. Gudehus et al. first studied the picking process as an optimisation goal and proposed a preliminary framework of a manual picking system [1]. This work lays the foundation for the subsequent person-to-goods picking system. Subsequent scholars have expanded research on the order batching problem across multiple dimensions, including the foundational order batching model, integrated order batching and sorting, joint optimisation of order batching and path planning, online dynamic order batching, multi-picker collaboration problem, complex warehouse layout and automation, multi-objective optimisation and cost control [2–8]. In person-to-goods picking systems, Pardo et al. review solution algorithms for the order batching problem across four distinct scenarios: offline single-picker, offline multi-picker, online single-picker and online multi-picker [9–13]. Driven by the growing demand for accelerated order fulfilment, goods-to-person systems are increasingly adopted in warehouse operations, primarily due to their ability to enhance throughput. In the study of the order batching problem, there are significant differences between goods-to-person picking systems and person-to-goods picking systems. Regarding optimisation objectives, goods-to-person picking systems aim to optimise robotic task allocation and path planning, whereas person-to-goods picking systems focus on minimising picks' walking paths. When considering constraints, goods-to-person picking systems must consider robot fleet size, power consumption, task load and moving speed, whereas person-to-goods picking systems are primarily constrained by warehouse spatial configuration and picker capacity. Based on this, the research on order batching under goods-to-person picking systems has become an important research hotspot in the field of modern logistics, which has attracted wide attention from academia and industry.

For goods-to-person picking systems, researchers have developed various optimisation methodologies, with exact algorithms and heuristic-based approaches being the most prevalent implementations. Exact algorithms can obtain the theoretically optimal solution [14, 15]. However, their computational complexity increases exponentially with problem size, severely limiting the solution efficiency and rendering them unsuitable for large-scale order scenarios. Heuristic-based approaches encompass methodologies, such as a heuristic method combining TSP and "S" shelf strategy, a heuristic method combining genetic algorithm and adaptive neighbourhood search, etc [16, 17]. While these algorithms enhance the timeliness, they exhibit critical limitations, including high parameter sensitivity and poor scalability when addressing large-scale order batching problems. The core of the order batching problem is to efficiently cluster the order set. Therefore, the clustering method can more accurately fit for solving the problem. K-means is a commonly used clustering algorithm [18]. However, when the original K-means clustering algorithm deals with high-dimensional data scenarios, its clustering performance will be significantly reduced due to the sparsity of the feature space and the failure of the distance measure caused by the "curse of dimensionality". The pharmaceutical order data usually have multi-dimensional attributes, which seriously affect the practical application of the K-means clustering algorithm in the pharmaceutical field.

Based on the high dimensionality, sparsity and discreteness characteristics of pharmaceutical orders, the study introduces an order batching method that combines feature subset and clustering optimisation. Through dimensionality reduction and feature selection, the "curse of dimensionality" problem is effectively alleviated, and the K-means algorithm is improved to improve the stability and effectiveness of clustering results. Through feature optimisation, clustering algorithm improvement and scene adaptation, the improved K-means clustering algorithm based on EIQ analysis for order batching successfully solves the limitations of original methods in pharmaceutical order processing, improves stability, accuracy and processing efficiency, and provides an efficient solution for SBS/RS. Finally, this work conducts a case analysis based on the real order data of Company A for 30 days, and the performance differences between the algorithm proposed in this paper and the FCFS strategy and the original K-means algorithm are compared. Sensitivity analyses of the feature selection threshold and feature removal threshold are conducted to assess the impact of various parameter combinations. This evaluation provides a scientific basis for optimising algorithm parameter settings in practical applications. The applicability of the improved K-means algorithm proposed in this study is analysed, and the algorithm is extended to different pharmaceutical logistics companies. This method provides an innovative solution for the order batching problem in the SBS/RS, which has strong theoretical significance and practical value. It can be promoted in a wider range of industrial applications in the future, and provides support for further improving the efficiency and reliability of the intelligent warehousing system.

The structure of this paper is shown as follows. Section 2 summarises the relevant literature and analyses the advantages and disadvantages of the existing order batching algorithm. Section 3 introduces the improved K-means clustering algorithm based on EIQ analysis for order batching in detail. Section 4 verifies the effectiveness of the algorithm through case analysis and conducts parameter sensitivity analysis and algorithm applicability analysis. Section 5 summarises the full text and looks forward to the future research direction.

2. LITERATURE REVIEW

Order batching is the core optimisation link of the picking system, and its algorithm design directly affects picking efficiency and operation cost. For the order batching problem, the academic community has proposed multiple optimisation methods. For the person-to-goods picking system, Gil-Borrás et al. used a multi-start search and a variable neighbourhood descent algorithm to achieve order batching and load balancing [19]. Chen et al. combined association rule mining with 0-1 integer programming to construct an ARIP method, which effectively reduced the length of the picking path by maximising the correlation of orders in batches [20]. Pardo et al. summarised and refined the algorithms used in different scenarios, which can be summarised into four categories: mathematical optimisation algorithms, heuristic algorithms, meta-heuristic algorithms and hybrid algorithms [9]. Mathematical optimisation algorithms include integer linear programming, mixed integer programming, column generation, dynamic programming and branch and bound, which are mainly used for small-scale problems to find the optimal solution; heuristic algorithms include seed method, saving algorithm and S-type path strategy, which can quickly generate approximate solutions through simple rules [14, 21, 22]. Meta-heuristic algorithms such as genetic algorithm, simulated annealing, tabu search, variable neighbourhood search, ant colony optimisation, particle swarm optimisation are suitable for large-scale problems and can efficiently explore the solution space [23]. The hybrid algorithm combines the advantages of different algorithms, such as the combination of tabu search and genetic algorithm, mixed integer programming and heuristics, adaptive large neighbourhood search, deep reinforcement learning and quantum computing, which further improves the performance and applicability of the algorithm [24].

Due to the difference between the person-to-goods picking system and the goods-to-person picking system, the algorithm in the person-to-goods picking system usually assumes that the storage is fixed, while the goods-to-person picking system needs to process multiple automatic guided vehicle system (AGVS) scheduling and dynamic adjustment of storage bits in real time. Traditional integer programming or genetic algorithm is difficult to cope with second-level response requirements; moreover, the algorithms of the person-to-goods picking system usually focus on the path optimisation of a single picker, and the person-to-goods picking system needs to coordinate multiple links, such as automatic guided vehicle (AGV) handling. Direct application will lead to AGV path conflicts or idle workstations. Similarly, the algorithm design can be classified into three categories: mathematical optimisation algorithm, heuristic algorithm and hybrid algorithm. In the application scenario of a mathematical optimisation algorithm, Yang et al. constructed an integer linear programming clustering model for the AGVS, which effectively reduced the number of robot movements [25]. For the automated storage and retrieval system (AS/RS), a discrete-time mixed integer linear programming model and a preprocessing program are proposed by Zhao et al., which effectively reduces the completion time of the multi-purpose batch production system [26]. However, as the scale of the problem increases, the model construction time will increase exponentially.

In the application scenario of heuristic algorithm, a storage allocation model based on product similarity and a variable neighbourhood search algorithm combined with order alienation are proposed by Xiang et al. for the autonomous mobile robot system (AMRS) [27]. Nicolas et al. developed a simulated annealing algorithm and an integer linear programming model for the AS/RS [28]. Jiang et al. proposed a two-stage order batching model and a dynamic clustering algorithm, which effectively reduced the number of shelf movements in the AMRS and balanced the workload of the picking workstation [29]. However, only considering that each item is only stored on one shelf, it may be different from the actual multi-shelf storage. The ant colony optimisation algorithm is used by Hu et al. based on the design of AS/RS [30]. Through the double-layer genetic algorithm and BFD algorithm, Lei et al. effectively improved the outbound efficiency and warehouse utilisation of the AS/RS [31]. An improved Pareto optimal elite non-dominated sorting genetic algorithm is also proposed by Wang et al. for the AS/RS [32].

In the application scenario of a hybrid algorithm, Winkelhaus et al. proposed an intelligent optimisation algorithm (agent-based simulation model), which effectively evaluated the performance of a hybrid picking system in improving warehouse picking efficiency and reducing costs [33]. For the AGVS, Liang et al.

developed a hybrid time window strategy and an improved ant colony algorithm [34]. Xie et al. proposed a variable neighbourhood search algorithm and a two-commodity network flow formula [35]. Kucuksari et al. proposed a Lagrangian relaxation method and a simulated annealing algorithm based on K-means clustering [36]. However, the computational complexity of the model is high, and the solution time for large-scale instances is long. Bansal et al. proposed a two-stage order batching model and a dynamic clustering algorithm based on the AS/RS, which effectively reduced the number of shelf movements and balanced the workload of the picking workstation, but the stability of the model is low [37]. Wang et al. simulated an annealing algorithm and an integer linear programming model for AS/RS [38]. Zhen et al. developed a two-layer rotation algorithm to solve the order, robot and shelf scheduling problems in the AMRS, which effectively reduces the order completion time, but may reduce the optimisation effect when the order dispatch frequency is too high [39].

In summary, although mathematical optimisation algorithms and heuristic algorithms have achieved high-quality approximate solutions in a reasonable time by simulating natural phenomena or employing local search, they have shown obvious limitations in relying on initial conditions, parameter sensitivity and handling large-scale problems. Furthermore, various algorithms do not exhibit significant advantages over one another. Considering that the essence of the order batching problem is the optimal grouping of the order set, the clustering algorithm is designed to better realise the matching between the problem characteristics and the solution strategy. Through the effective aggregation of the orders, the processing efficiency and stability in large-scale scenarios are improved. When the original K-means clustering algorithm is directly applied to deal with high-dimensional data such as pharmaceutical orders, the clustering performance will be significantly reduced due to the "curse of dimensionality". The improved algorithms based on K-means clustering (such as the simulated annealing algorithm based on K-means clustering and the double-layer genetic algorithm based on K-means clustering) still have the problem of low computational efficiency in the face of large-scale examples.

Therefore, we present an improved K-means clustering algorithm based on EIQ analysis for order batching. The main contributions of this article are shown as follows. First, by using IK frequency filtering and Pearson correlation to remove redundancy, a low-dimensional and highly representative feature subset is constructed to address the "curse of dimensionality" problem. Second, the algorithm optimises cluster centre selection with a roulette strategy and enhances order similarity within batches using cosine distance.

3. METHODOLOGY

This study establishes an integrated methodological framework to address the order batching challenges in the pharmaceutical warehouse SBS/RS. The proposed approach systematically addresses three key challenges: (1) the contradiction between high-dimensional SKU representations and computational tractability; (2) the mismatch between sparse order characteristics and conventional similarity metrics; (3) the instability of original clustering algorithms in pharmaceutical logistics scenarios. By synergising feature engineering with algorithmic innovation, the methodology achieves significant improvements in reducing total bin presentations while maintaining strict compliance with pharmaceutical storage regulations.

3.1 Problem description

This paper studies the order batching problem in SBS/RS in the context of pharmaceutical logistics warehousing. The aim is to optimise order batching to minimise total bin presentations. Total bin presentations dispatched from the warehouse can, to a certain extent, approximate the processing time or equipment utilisation rate. Therefore, to simplify, we use total bin presentations to approximately reflect the overall performance. The problem assumes that all order information is fully known and fixed before batching, and the fact that the pharmaceutical warehouse inventory always meets the order demand. The objective is to reasonably divide these orders into several batches to minimise total bin presentations. Key challenges include: due to batch production management, each bin can only store a single SKU without mixing; multiple orders often have highly overlapping SKU demands, so assigning them to the same batch requires only one bin presentation for all related orders; the problem exhibits high dimensionality (due to many SKU types causing the "curse of dimensionality"), sparsity (orders contain only a few SKUs, invalidating traditional distance metrics) and discreteness (no natural order among SKUs), posing significant algorithm design challenges. A reasonable batching method maximises SKU demand overlap within the same batch, significantly reducing bin presentations and improving overall picking efficiency. Basic constraints include that all order lines of the

same order must be assigned to the same batch (no splitting), and the number of orders per batch cannot exceed a preset maximum.

Furthermore, the traditional K-means algorithm is highly sensitive to initial cluster centre selection, prone to local optima, and results in unstable clustering. These challenges are especially prominent in pharmaceutical logistics, where increasing SKU variety and order complexity make traditional batching methods inadequate. Therefore, a novel algorithm integrating feature selection and clustering optimisation is urgently needed to address these issues and meet the specific requirements of pharmaceutical logistics.

3.2 Improved K-means clustering algorithm based on EIQ analysis for order batching

This paper develops an improved K-means clustering algorithm based on EIQ analysis for order batching, redefining feature representation and clustering mechanisms for pharmaceutical order batching. The innovation lies in three interconnected components: a dual-stage feature selection mechanism, probabilistic cluster centres selection and directional similarity measurement.

Overall algorithm framework

The study develops an improved K-means clustering algorithm within the EIQ analysis framework for order batching. First, for the input order data set, based on the EIQ analysis framework, one-hot encoding is used to map the orders into high-dimensional feature vectors. Then, a two-stage feature selection is performed: important features are selected based on IK frequency analysis, and redundant features are removed using the Pearson correlation coefficient, constructing a low-dimensional representative feature subset to alleviate the "curse of dimensionality" and enhance feature representativeness. Second, to address the sensitivity of traditional K-means to initial cluster centres and the limitations of Euclidean distance in high-dimensional sparse data, an improved strategy is proposed. On one hand, a roulette wheel selection strategy is used to optimise the selection of cluster centres, reducing the risk of falling into local optima. On the other hand, cosine distance is adopted instead of Euclidean distance to measure the directional similarity and SKU overlap between orders, thereby more accurately grouping orders with similar demand into the same batch. The specific flow chart of the algorithm is shown in *Figure 1*.

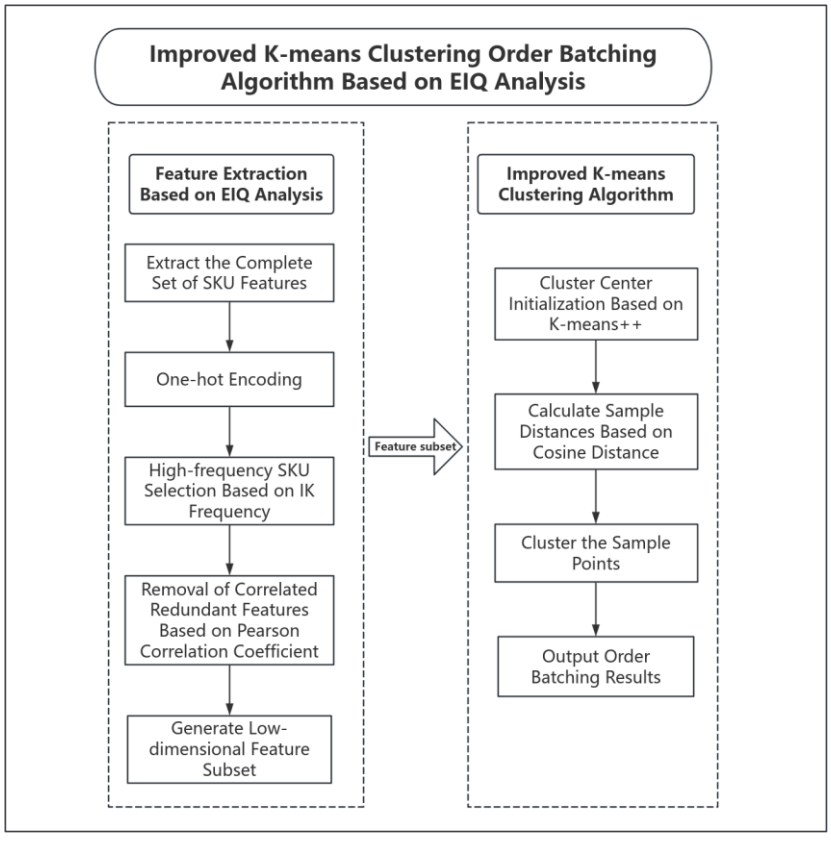

Figure 1 – Improved K-means clustering algorithm based on EIQ analysis for order batching flowchart

Feature extraction based on one-hot encoding

Considering the high-dimensional, sparse and discrete characteristics of pharmaceutical orders, one-hot encoding is well suited to mapping each order onto a 0/1 vector across several possible SKU dimensions. The process is as follows:

1) Extract the complete set of SKU features

For the order set $O=\{o_1,o_2,\ldots,o_i\ldots,o_N\}$, all distinct SKUs involved are recorded $\{SKU_1,SKU_2,\ldots,SKU_j\ldots,SKU_M\}$. However, before any screening, there may theoretically be thousands or even tens of thousands of SKUs, requiring further "feature selection" thereafter.

2) One-hot encoding

For any given order o_i, if it contains SKU_j, then the vector x_i at dimension is set to 1; otherwise, it is 0. Thus, each order is encoded as a vector:

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, \ldots, x_{ij} \ldots, x_{iM}), x_{ij} \in \{0,1\} \tag{1}$$

Here, the dimension M may be very large, so feature selection is essential to reduce dimensionality and mitigate the "curse of dimensionality".

Feature selection based on IK frequency and Pearson correlation coefficient

To improve clustering performance, it is necessary to further filter the one-hot encoded features by removing unimportant or redundant features, thereby forming a smaller and more optimal feature subset. This process consists of two parts: important feature analysis and feature correlation analysis.

1) Important feature analysis based on IK frequency

By calculating the order frequency ik_j of each SKU, get the number of orders in which the SKU appears. Sort all SKUs in descending order according to their frequencies ik_j:

$$\boldsymbol{ik_{(1)}} \geq \boldsymbol{ik_{(2)}} \geq \cdots \geq \boldsymbol{ik_{(M)}} \tag{2}$$

Set a feature extraction threshold n, and select the top n% SKUs as the feature subset to construct the feature vectors of orders.

When n is too small, the feature subset is insufficient to accurately represent differences between orders; when n is too large, it leads to the "curse of dimensionality" and noise interference. Select the value or range of n at which the total bin presentations stabilise with respect to changes in n, and the clustering performance is optimal.

2) Feature correlation analysis based on the Pearson correlation coefficient

Even after selecting the top n SKUs, some features may still be highly redundant, providing almost identical "explanations" for the same batch of orders. To further remove such redundancy, the Pearson correlation coefficient is used to measure the linear correlation between features:

For any two features f_p and f_q , their corresponding order sample vectors are:

$$\boldsymbol{f_p} = (f_p(o_1), f_p(o_2), \ldots, f_p(o_i), \ldots, f_p(o_N)) \tag{3}$$

$$\boldsymbol{f_q} = (f_q(o_1), f_q(o_2), \ldots, f_q(o_i), \ldots, f_q(o_N)) \tag{4}$$

where $f_p(o_i)=1$, if order o_i contains feature $\boldsymbol{f_p}$, otherwise 0.

The Pearson correlation coefficient between features $\boldsymbol{f_p}$ and $\boldsymbol{f_q}$ is calculated as:

$$r_{p.q} = \frac{\sum_{i=1}^{N}(f_p(o_i) - \bar{f}_p)(f_q(o_i) - \bar{f}_q)}{\sqrt{\sum_{i=1}^{N}(f_p(o_i) - \bar{f}_p)^2 \sum_{i=1}^{N}(f_q(o_i) - \bar{f}_q)^2}} \tag{5}$$

where $\bar{f}_p$ and $\bar{f}_q$ are the mean values of the vectors $\boldsymbol{f_p}$ and $\boldsymbol{f_q}$, respectively.

Set a feature removal threshold m. If $r_{p,q}>m$, it indicates a high positive correlation between the two features on the order dataset. In this case, retain the feature with the higher IK frequency and remove the other.

Repeat this process until the Pearson correlation coefficient between any two retained features does not exceed m.

Through the above important feature analysis and correlation analysis, a more representative and less redundant feature subset can be selected from the massive SKU set, providing an efficient basis for subsequent clustering.

Improved K-means clustering

After selecting the feature subset, each order (sample) can be represented as a relatively low-dimensional binary feature vector:

$$\boldsymbol{x_i} = (x_{i1}, x_{i2}, \dots, x_{ij} \dots, x_{id}), x_{ij} \in \{0,1\} \tag{6}$$

where d is much smaller than the original dimension M. At this point, the improved K-means algorithm can be applied for clustering and batching.

1) Cluster centre initialisation based on the roulette wheel selection strategy

To reduce the sensitivity of the algorithm to the initial cluster centres and avoid falling into local optima, the roulette wheel selection strategy is adopted for selecting cluster centres. The steps are as follows:

Step 1: Randomly select one sample C_1 as the first centre;

Step 2: For each sample x not yet chosen as a centre, compute its minimum distance to all currently selected centres and record it as $D(x)$;

Step 3: Let the probability of the next centre being selected be:

$$P(x = C_{next}) = \frac{D(x)^2}{\sum_{z \in \chi} D(z)^2} \tag{7}$$

and use the roulette wheel method to draw C_{next};

Step 4: Repeat until K initial centres are chosen.

2) Similarity measurement based on cosine distance

Euclidean distance measures the absolute distance between points in space, while cosine similarity measures the angle between vectors. Given that the feature vectors are binary and sparse, emphasising the difference in feature direction rather than magnitude, cosine similarity better captures the overlap of SKU demands between orders. Cosine distance, derived from cosine similarity, is thus used as the distance metric.

For two order feature vectors $x=(x_1,x_2,\dots,x_d)$ and $y=(y_1,y_2,\dots,y_d)$, the Euclidean distance $d_E(x,y)$ is:

$$d_E(x,y) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \tag{8}$$

Cosine similarity is:

$$\cos(x,y) = \frac{x \cdot y}{\|x\| \cdot \|y\|} \tag{9}$$

where $||x||$ and $||y||$ are the Euclidean norms of x and y, respectively.

$$\|x\| = \sqrt{\sum_{i=1}^{d} x_i^2} \tag{10}$$

$$\|y\| = \sqrt{\sum_{i=1}^{d} y_i^2} \tag{11}$$

Cosine similarity ranges from −1 to 1, with values closer to 1 indicating higher similarity.

The cosine distance is defined as:

$$d^c(x,y) = 1 - \cos(x,y) \tag{12}$$

ranging from 0 (identical vectors) to 2 (completely opposite vectors).
The pseudocode of the improved K-means clustering algorithm is shown in *Table 1*.

Table 1 – Improved K-means clustering algorithm pseudo code

Algorithm: Improved K-means (based on Cosine Distance)	
Input: Sample matrix X, number of clusters K, maximum iterations max_iter,number of initialisations n_init	
Output: Best clustering labels, cluster centres	
1	function ImprovedKMeans(X: matrix[N,d], K: int, max_iter: int,
2	n_init: int) -> (labels: vector[N], centres: matrix[K,d])
3	best_inertia ← +∞
4	best_labels ← Ø
5	best_centres ← Ø
6	for run in 1 … n_init do
7	centres ← InitializeCentres(X, K)
8	for iter in 1 … max_iter do
9	labels ← AssignLabels(X, centres)
10	new_centres ← UpdateCentres(X, labels, K)
11	if Converged(centres, new_centres) then
12	break
13	centres ← new_centres
14	end for
15	inertia ← ComputeInertia(X, labels, centres)
16	if inertia < best_inertia then
17	best_inertia ← inertia
18	best_labels ← labels
19	best_centers ← centres
20	end if
21	end for
22	return best_labels, best_centres
23	end function
24	function InitializeCentres(X: matrix[N,d], K: int) -> matrix[K,d]
25	centres ← [RandomSample(X)]
26	while |centres| < K do
27	D2 ← [MinCosineDistance²(x, centres) for x in X]
28	p ← Normalise(D2)
29	centres.append(RouletteWheelSample(X, p))
30	end while
31	return L2Normalise(centres)
32	end function

```
function AssignLabels(X: matrix[N,d], centres: matrix[K,d]) -> vector[N]
    labels ← vector[N]
    for i in 1 … N do
        distances ← [ CosineDistance(X[i], centres[j]) for j in 1 … K ]
        labels[i] ← argmin(distances)
    end for
    return labels
end function
function UpdateCentres(X: matrix[N,d], labels: vector[N], K: int) -> matrix[K,d]
    centres ← matrix[K,d]
    for k in 1 … K do
        cluster ← { X[i] | labels[i] = k }
        if cluster ≠ Ø then
            centres[k] ← L2Normalise( Mean(cluster) )
        else
            centres[k] ← RandomSample(X)
        end if
    end for
    return centres
end function
function ComputeInertia(X: matrix[N,d], labels: vector[N], centres: matrix[K,d]) -> float
    inertia ← 0
    for i in 1 … N do
        inertia ← inertia + CosineDistance(X[i], centres[ labels[i] ])
    end for
    return inertia
end function
function CosineDistance(a: vector[d], b: vector[d]) -> float
    return 1 − ( a·b ) / ( ||a|| · ||b|| )
end function
```

3.3 Algorithm time complexity analysis

This section starts from the two aspects of feature selection and clustering, compares it with the original K-means, and explains the time complexity of the improved algorithm.

Suppose the number of orders (samples) is N, the initial SKU dimension is S, the dimension retained after two-stage feature selection is d ($d \ll S$), the number of clusters is K, the maximum number of iterations of an initialisation process is I, and the number of random initialisations to be performed to avoid local optimality is n_{init}.

In the feature selection phase, N orders are first one-hot encoded and the frequency of each SKU in the sample is counted. The time complexity of this process is $O(NS)$. Then, all SKU frequencies are sorted, which takes $O(S\log S)$. After the frequency sorting is completed, the algorithm calculates the pairwise Pearson correlation coefficient on the reduced candidate set to remove redundant features. The complexity is $O(d^2N)$. Since d is the number of dimensions retained after screening, and d is much smaller than S, this item does not become a bottleneck. In summary, the overall complexity of the feature selection phase is approximately $O(NS)$, and this step only needs to be performed once, which has a limited impact on the subsequent calculation amount.

The core time consumption of the clustering phase comes from the repeated "initialisation-iteration" process. The roulette strategy is used to select the initial centre, which requires traversing all samples, with a complexity of $O(KN)$. In each subsequent iteration, the algorithm calculates the cosine distance between each sample and K centres and updates the cluster centre vector. The combined complexity of the two is approximately $O(NKd)$. If a single initialisation iteration takes at most I rounds to converge, the time consumption for one initialisation is $O(INKd)$; plus n_{init} random restarts, the total time consumption is $O(n_{init}INKd)$. Compared with the complexity $O(n_{init}INKS)$ of traditional K-means running directly on the full-dimensional space S, the improved algorithm can theoretically achieve an acceleration of about S/d times.

To sum up, the overall time complexity of the algorithm can be written as:

$$T(N) = O(NS) + O(n_{init}INKd) \tag{13}$$

4. CASE STUDY

We conduct a comprehensive empirical validation of the proposed algorithm using real-world pharmaceutical logistics data. The case study systematically evaluates algorithm performance across three dimensions: parameter sensitivity, operational efficiency and scenario adaptability, providing actionable insights for warehouse optimisation.

4.1 System parameter settings

This study uses real data from Company A for analysing the case. Company A's distribution centre employs SBS/RS for order picking. Each order requires extracting SKUs from bins to the order boxes on the picking station. To improve picking efficiency, multiple orders are grouped into several batches. Company A's current order processing mode adopts the FCFS strategy, accumulating orders into one batch of 200. One order can have multiple order lines, each corresponding to a SKU. Here, a SKU is defined as a combination of (product code, batch production), meaning the SKU is uniquely identified by both product code and batch production. The core objective is to minimise total bin presentations, thereby maximising picking efficiency.

4.2 Results analysis

This paper uses 30 days of order data from Company A as 30 test datasets. *Figure 2* displays the daily order quantities over 30 days, which reflect fluctuations in order volumes. The performance of the three methods is analysed under the parameter settings of n=25 and m=0.8. Specifically, the three methods include: FCFS, unimproved K-means clustering and improved K-means clustering based on EIQ analysis. The unimproved K-means algorithm employs randomly selected cluster centres and Euclidean distance without feature subset screening, while the improved K-means conducts feature extraction, adopts n% candidate features, eliminates highly correlated features and utilises the roulette wheel method to select cluster centres with cosine distance as the distance metric.

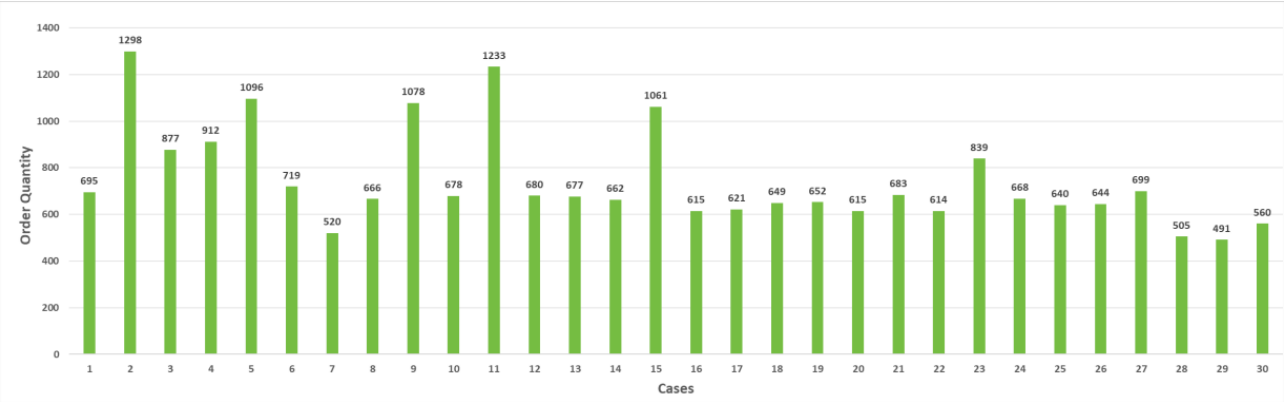

Figure 2 – 30 sets of test data order quantity

Figure 3 shows the comparison of total bin presentations of three methods (FCFS, original K-means and improved K-means) within 30 days. The experimental results show that the FCFS method has the highest total bin presentations, and the improved K-means algorithm performs best, with the total bin presentations reduced

by up to 18% compared with FCFS and up to 13% compared with the original K-means algorithm. In the 30 tests, the improved K-means algorithm always performed best, with the total bin presentations significantly lower than FCFS and the original K-means algorithm.

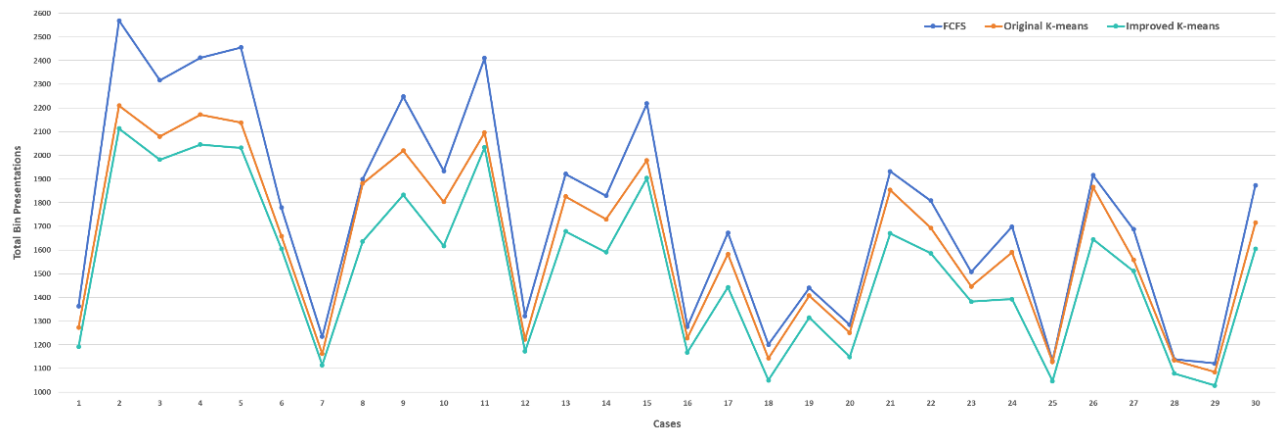

Figure 3 – Results comparison chart

As shown in *Figure 4*, after testing 30 sets of real enterprise order data, the solution time of the improved K-means algorithm was controlled between 6 s and 25 s, with an average of about 13 s and a maximum of only 25 s, all of which were far below the 50 s upper limit allowed by the project. This shows that the algorithm can reliably complete the calculation within the specified time, which not only meets the real-time business requirements but also fully proves its feasibility in the production environment.

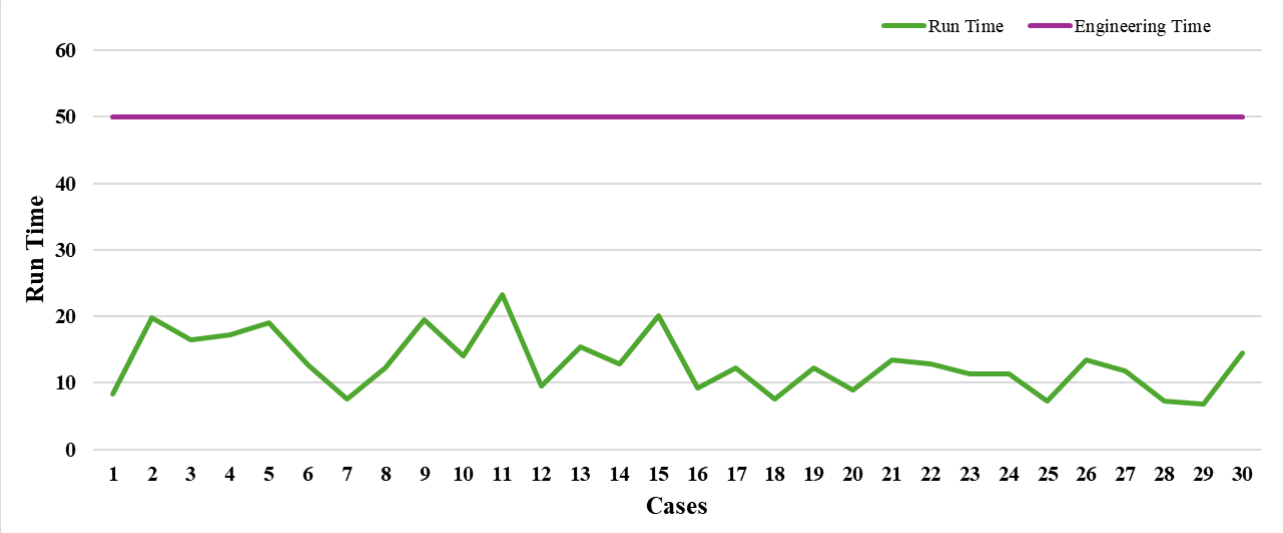

Figure 4 – Algorithm feasibility diagram

Figure 5 shows the comparison of the total bin presentations of the three methods after 100 repeated runs on the same data set. The results show that the FCFS strategy has the same results after multiple runs due to fixed order data, without any fluctuations. The original K-means algorithm has significant fluctuations in results due to the random selection of cluster centres, and its coefficient of variation is 0.557. The improved K-means algorithm optimises the selection of cluster centres through the roulette strategy and uses the cosine distance to measure the similarity of orders, which effectively reduces the impact of randomness, reduces the coefficient of variation to 0.279, and significantly reduces volatility. The improved K-means algorithm shows higher stability and consistency in multiple repeated runs.

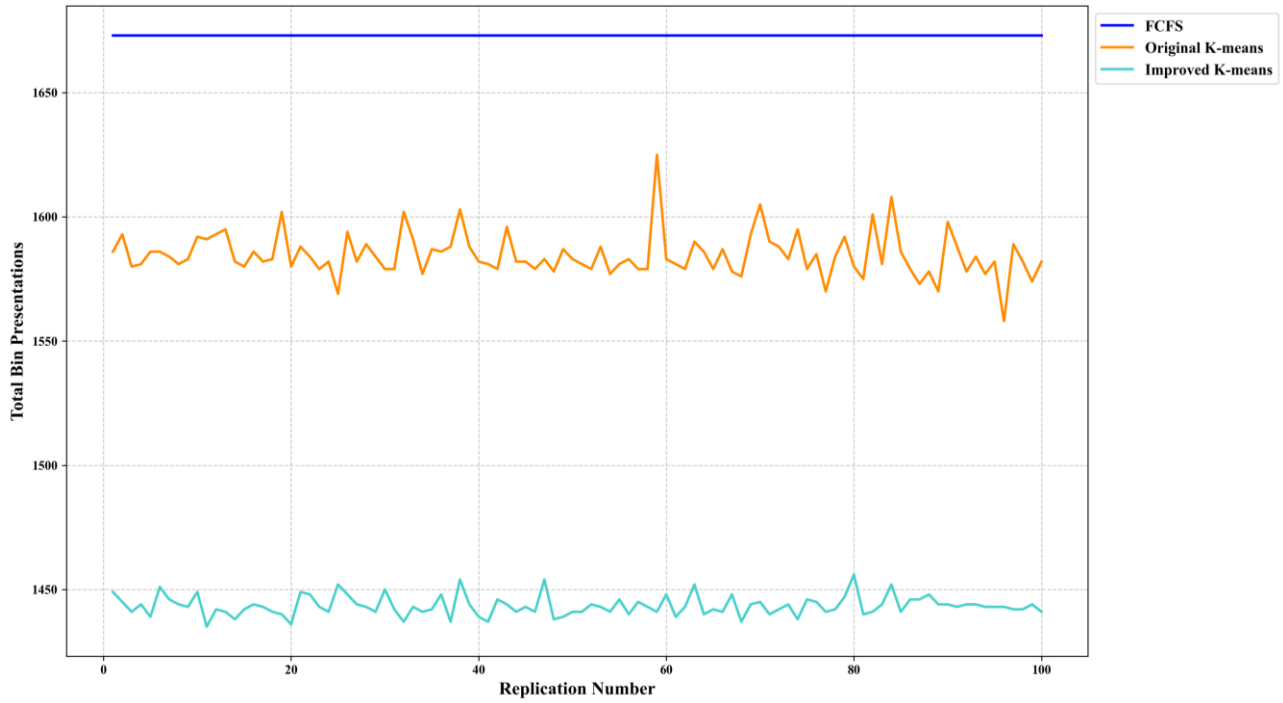

Figure 5 – Algorithm stability comparison chart

To further verify the stability of the algorithm, we define the SKU popularity p_j as the ratio of the order frequency of a certain SKU to the total number of orders, that is:

$$p_j = \frac{IK_j}{N} \tag{14}$$

It is used to measure the demand intensity of a single SKU, where IK_j represents the order frequency of SKU j, and N is the total number of orders in the dataset.

We can further obtain the key indicator for measuring the complexity of the order structure, namely, the single-row ratio of order data:

$$\sum_{j=1}^{M} p_j = \sum_{j=1}^{M} \frac{IK_j}{N} = \frac{1}{N} \sum_{j=1}^{M} IK_j \tag{15}$$

where $\sum_{j=1}^{M} IK_j$ is the sum of all order lines (each order line corresponds to one SKU) and N is the total number of orders.

On this basis, refer to the Herfindahl-Hirschman Index (HHI, a comprehensive index to measure industry concentration);

$$HHI = \sum_{i=1}^{n} (\frac{x_i}{X})^2 \tag{16}$$

where x_i/X represents the market share of the ith enterprise. Obviously, the larger the HHI is, the higher the market concentration and the higher the degree of monopoly.

Considering the impact of order structure complexity on concentration, the popular concentration C of the data set is defined as

$$C = \frac{\sum_{j=1}^{M} p_i^2}{\sum_{j=1}^{M} p_j} \tag{17}$$

This allows C to more purely reflect the concentration or dispersion of the SKU popularity distribution.

This paper selects 8 datasets with a popularity concentration interval of [0.004, 0.0075] and a step size of 0.0005. The improved K-means algorithm is run independently on each dataset 100 times, and the total number of box outbound times of each run is recorded. The results are shown in *Figure 6*. For the results of multiple runs of the algorithm on different datasets, the distribution of the number of box outbound times shows a high degree of stability and consistency, which shows that when the popularity concentration of the data set changes, that is, the SKU popularity changes, the output results of the algorithm are highly repeatable and stable.

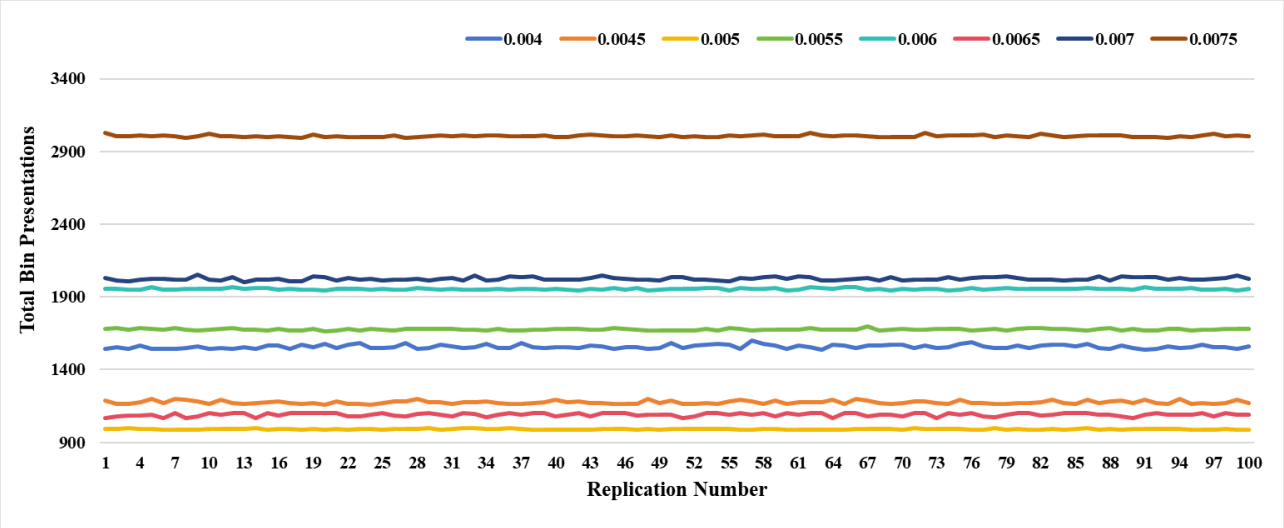

Figure 6 – The popularity concentration of the datasets is different

This work further selects multiple datasets, the popularity concentrations of which all fall within the interval [0.0050, 0.0055], and independently runs the improved K-means algorithm 100 times on each dataset, recording the total number of material box outbound trips for each run, and the results are shown in *Figure 7*. On these datasets with similar popularity concentrations, the total number of material box outbound trips obtained by multiple runs of the algorithm is highly similar, which shows that when the popularity concentration of the dataset remains basically unchanged, that is, the SKU popularity remains basically unchanged, the output results of the algorithm are still highly repeatable and stable.

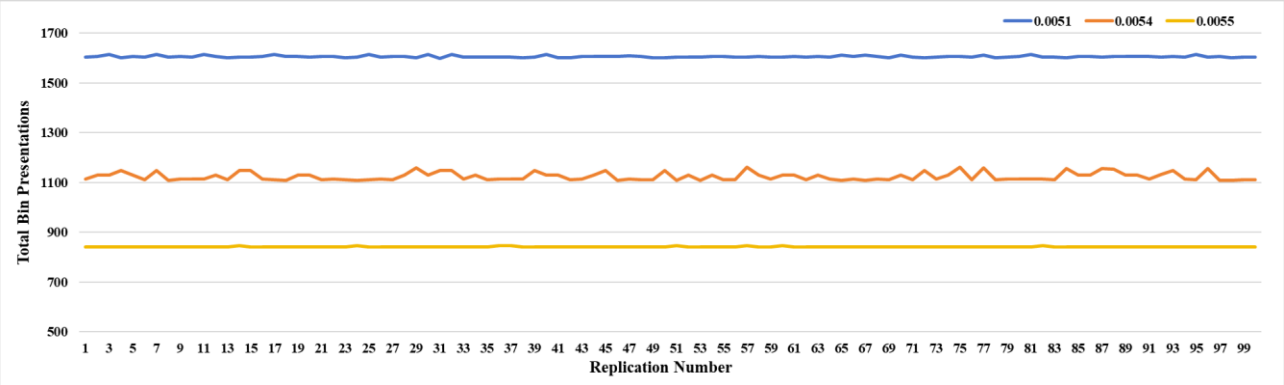

Figure 7 – The popularity concentration of the data set is within a fixed interval

Therefore, regardless of whether the SKU popularity changes, the proposed improved K-means algorithm can continue to output similar and reliable results, showing a high degree of stability.

4.3 Sensitivity analysis

Regarding the unlabelled order data from Company A, the silhouette coefficient is chosen as the core evaluation metric for clustering quality. This metric quantifies intra-cluster cohesion and inter-cluster separation, providing a comprehensive assessment of clustering performance. For each sample i, the silhouette coefficient is calculated by comparing the average distance to other samples in the same cluster $a(i)$ and to the nearest other cluster $b(i)$, with the global silhouette coefficient being the average over all samples.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{18}$$

The coefficient ranges from −1 to 1, where values closer to 1 indicate better cluster structure.

Using order data from one day of Company A, a sensitivity analysis is conducted on the feature selection threshold n and feature removal threshold m. The parameter n is varied from 5 to 100 (step 10), and m from 0.6 to 1.0 (step 0.1). A grid search over all parameter combinations records total bin presentations and silhouette coefficients. The ideal parameters minimise total bin presentations and maximise silhouette coefficients. The 3D heatmap in *Figure 8* illustrates the optimal ranges for n and m.

The experimental results show that when the feature selection threshold n is between 20 and 25 and the feature removal threshold m is between 0.8 and 0.9, the algorithm performance is optimal: the total bin presentations are reduced to the minimum (1,136 times), and the silhouette coefficient is increased to 0.6, indicating that the clustering effect is at a high level in terms of intra-cluster compactness and inter-cluster separation. n is within this range, which just reflects that the top 20% of the drugs in the IK ranking can cover about 80% of the key indicators, which is in line with the theory of "20% of Class A goods account for 80 effect" in ABC classification; at the same time, m between 0.8 and 0.9 can screen out extremely strong correlations, ensure the consistency within the cluster and effectively eliminate redundant data.

It is recommended that companies prioritise setting the feature selection threshold n to 20-25 and the feature removal threshold m to 0.8-0.9 to optimise algorithm performance and reduce the risk of conveyor line congestion; at the same time, integrate the algorithm into the warehouse management system (WMS) to realise automated batching, regularly and dynamically update SKU features, adapt to business fluctuations, and support the efficient operation of the pharmaceutical distribution business.

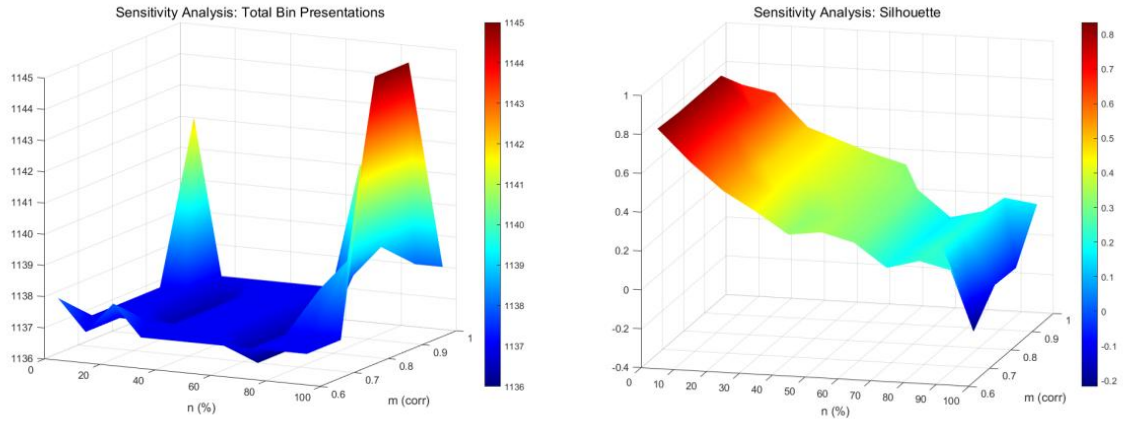

Figure 8 – Sensitivity analysis chart

4.4 Algorithm applicability analysis

This study selects data from different order structures to compare and analyse the applicability of the algorithm. The horizontal axis is the item lines per order, which range from [2.0, 5.0] with a step size of 0.5, and the vertical axis is the total bin presentations.

The comparative analysis in *Figure 9* shows that the improved K-means algorithm is always better than FCFS and the original K-means algorithm under different item lines per order. When the item lines per order are $\leqslant$ 3.5, the improved algorithm can reduce the total bin presentations by about 3% at most compared with the unimproved K-means; and when the item lines per order are >3.5, the optimisation effect is further expanded to 6%. As the item lines per order increase, the optimisation effect of the improved k-means algorithm becomes more obvious.

The improved K-means algorithm can effectively reduce the total bin presentations and improve efficiency under different order structures, and is particularly suitable for complex order scenarios with high item lines per order; while the FCFS algorithm has poor applicability under complex order structures and is not recommended for use in high item lines per order scenarios. For different pharmaceutical logistics companies, the improved K-means algorithm can be considered to optimise the total bin presentations and improve operational efficiency.

In industries such as e-commerce B2C, auto parts and book publishing, the single-row ratio of order data is in the range of [3, 4.5], and the order data are generally high-dimensional, sparse and discrete, which is consistent with the characteristics of the improved K-means algorithm. As can be seen from the figure, the algorithm proposed in this paper can improve operating efficiency under different order structures. Therefore, for the industries mentioned above, this algorithm can be directly integrated into the existing picking system, allowing it to deliver its benefits and demonstrating its feasibility and value for cross-industry promotion.

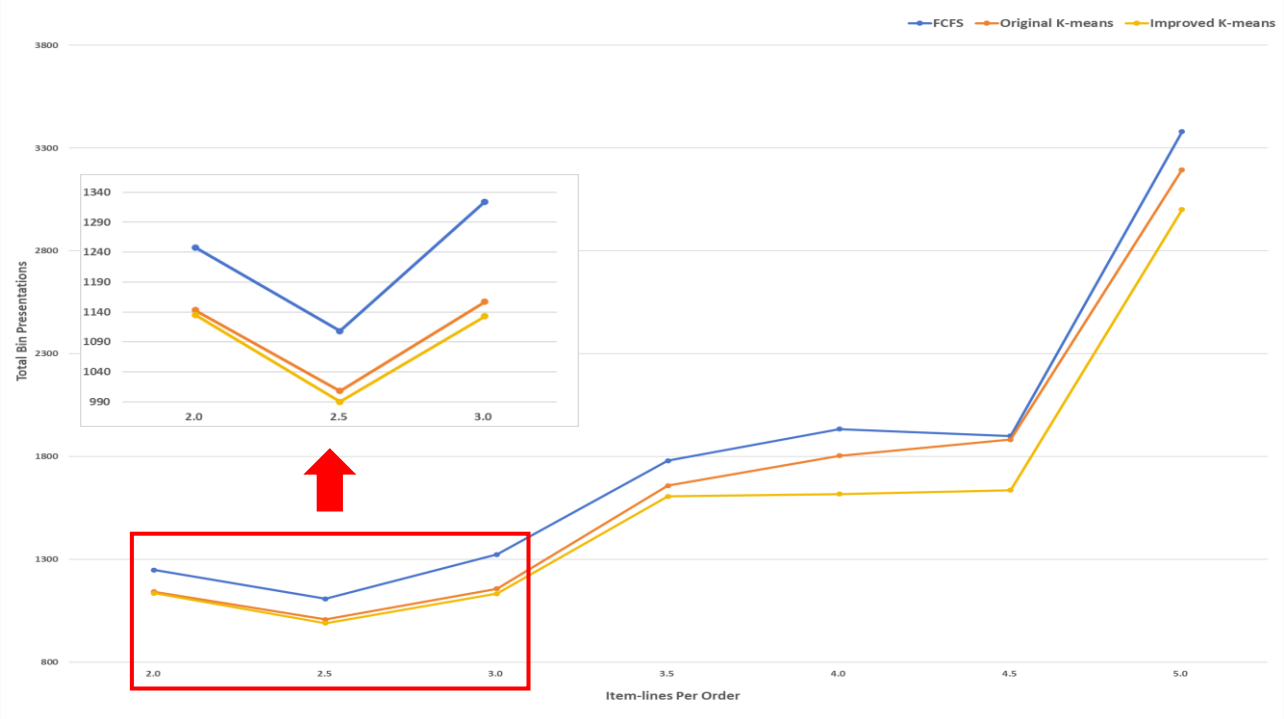

Figure 9 – Algorithm suitability analysis chart

5. CONCLUSION

This paper examines the order batching problem in SBS/RS and proposes an improved K-means clustering algorithm based on EIQ analysis. The algorithm addresses the issue of "curse of dimensionality" by constructing a low-dimensional, highly representative feature subset through IK frequency filtering and Pearson correlation-based redundancy elimination. A roulette wheel selection strategy is employed to optimise the cluster centres, and cosine similarity is used to enhance order similarity. Ultimately, the algorithm aims to minimise total bin presentations, effectively improving the order batching efficiency in SBS/RS.

Case studies and sensitivity experiments demonstrate that compared with the traditional FCFS strategy and the original K-means algorithm, the proposed method shows clear advantages on 30 days of real order data from Company A, with smaller fluctuations and higher stability in processing results. Regardless of whether the SKU popularity changes, the algorithm in this work can output similar and reliable results, showing a high degree of stability. Sensitivity analysis further confirms that the algorithm performs best when the feature selection threshold n is between 20 and 25 and the feature removal threshold m is between 0.8 and 0.9, consistent with the ABC classification theory of "20% A-items accounting for 80% effect", which has practical business significance. In addition, the algorithm applicability analysis shows that as the item lines per order increase, the optimisation effect of the improved K-means algorithm becomes more obvious, indicating that the algorithm in this article has practical value for different pharmaceutical logistics companies, and it has the feasibility and value of cross-industry promotion.

From a theoretical perspective, this study fills the gap in handling high-dimensional sparse data by synergistically optimising feature selection and clustering algorithms, providing a new approach for order batching problems in SBS/RS. Based on experience with integrating algorithms in previous projects, it takes about 2-3 weeks to integrate the algorithm into the existing warehouse management system (WMS), followed by 2-3 weeks of field testing to fully verify its effectiveness and stability. Practically, the algorithm has

significant engineering value and can be directly applied in SBS/RS to improve picking efficiency and reduce operational costs.

Future research may focus on two directions: first, developing online batching strategies for dynamic order scenarios combined with reinforcement learning to achieve real-time decision optimisation; second, expanding to a multi-objective optimisation framework that simultaneously considers total bin presentations, order timeliness and picking path efficiency.

ACKNOWLEDGEMENTS

This research is supported by the Enterprise Project (Grant # A2025076). We also wish to express our sincere gratitude to the following individuals for their invaluable contributions: Xun Weng provided the key insight that changes in SKU popularity could impact clustering stability. He then designed and conducted the corresponding validation experiments, which greatly enhanced the robustness of our findings. Hongxue Yang made significant contributions to the methodological design. Specifically, she proposed integrating IK frequency with Pearson correlation for feature selection. She also led the investigation into the algorithm's cross-industry applicability, demonstrating its potential for wider use.

REFERENCES

[1] Gudehus T. *Principles of order picking: Operations in distribution and warehousing systems*. Essen, Germany: W. Girardet Publishing; 1973.

[2] Elsayed EA. Algorithms for optimal material handling in automatic warehousing systems. *The International Journal of Production Research*. 1981;19(5): 525-535. DOI: 10.1080/00207548108956683.

[3] Henn S, et al. Metaheuristics for the order batching problem in manual order picking systems. *Business Research*. 2010;3:82-105. DOI: 10.1007/bf03342717.

[4] Gademann N, Velde S. Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE transactions*. 2005;37(1):63-75. DOI: 10.1080/07408170590516917.

[5] Tang L C, Chew E P. Order picking systems: batching and storage assignment strategies. *Computers & Industrial Engineering*. 1997;33(3-4): 817-820. DOI: 10.1016/S0360-8352(97)00245-3.

[6] De Koster R, Le-Duc T, Roodbergen KJ. Design and control of warehouse order picking: A literature review. *European journal of operational research*. 2007;182(2):481-501. DOI: 10.1016/j.ejor.2006.07.009.

[7] Schiffer M, et al. Optimal picking policies in e-commerce warehouses. *Management Science*. 2022;68(10):7497-7517. DOI: 10.1287/mnsc.2021.4275.

[8] Bukchin Y, Khmelnitsky E, Yakuel P. Optimizing a dynamic order-picking process. *European Journal of Operational Research*. 2012;219(2):335-346. DOI: 10.1016/j.ejor.2011.12.041.

[9] Pardo EG, et al. Order batching problems: Taxonomy and literature review. *European Journal of Operational Research*. 2024;313(1):1-24. DOI: 10.1016/j.ejor.2023.02.019.

[10] Hsu CM, Chen KY, Chen MC. Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in industry*. 2005;56(2):169-178. DOI: 10.1016/j.compind.2004.06.001.

[11] Matusiak M, et al. A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*. 2014;236(3):968-977. DOI: 10.1016/j.ejor.2013.06.001.

[12] Henn S. Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*. 2012;39(11):2549-2563. DOI: 10.1016/j.cor.2011.12.019.

[13] Zhang J, Wang X, Huang K. Integrated on-line scheduling of order batching and delivery under B2C e-commerce. *Computers & Industrial Engineering*. 2016;94:280-289. DOI: 10.1016/j.cie.2016.02.001.

[14] Muter I, Öncan T. An exact solution approach for the order batching problem. *Iie Transactions*. 2015;47(7):728-738. DOI: 10.1080/0740817X.2014.991478.

[15] Liu Z, Lu J, Ren C, et al. Joint optimization of storage assignment and order batching in robotic mobile fulfillment system with dynamic storage depth and surplus items. *Computers & Industrial Engineering,* 2025;200:110767. DOI: 10.1016/j.cie.2024.110767.

[16] Liu JE, Zhang S, Liu H. Research on AGV path planning under “parts-to-picker” mode. *Open Journal of Social Sciences*. 2019;7(6):1-14. DOI: 10.4236/jss.2019.76001.

[17] Feng J, et al. A bi-level optimization of storage allocation and shelf sequencing in a goods-to-person picking system. *Proceedings of the 5th International Conference on Computer Engineering and Application (ICCEA), 12-14 April. 2024, Hangzhou, China.* 2024. p. 1478-1486. DOI: 10.1109/ICCEA62105.2024.10603533.

[18] Lloyd S. Least squares quantization in PCM. *IEEE Transactions on Information Theory*. 1982;28(2):129-137. DOI: 10.1109/tit.1982.1056489.

[19] Gil-Borrás S, et al. A heuristic approach for the online order batching problem with multiple pickers. *Computers & Industrial Engineering*. 2021;160:107517. DOI: 10.1016/j.cie.2021.107517.

[20] Chen MC, Wu HP. An association-based clustering approach to order batching considering customer demand patterns. *Omega*. 2005;33(4):333-343. DOI: 10.1016/j.omega.2004.05.003.

[21] Cano JA, Correa-Espinal AA, Gómez-Montoya RA. Solución del problema de conformación de lotes en almacenes utilizando algoritmos genéticos. *Información tecnológica*. 2018;29(6):235-244. DOI: 10.4067/S0718-07642018000600235.

[22] Gibson DR, Sharp GP. Order batching procedures. *European journal of operational research*. 1992;58(1):57-67. DOI: 10.1016/0377-2217(92)90235-2.

[23] Gil-Borrás S, et al. Basic VNS for a variant of the online order batching problem. *Proceedings of the International Conference on Variable Neighborhood Search, 2019, Cham, Switzerland.* 2019. p. 17-36. DOI: 10.1007/978-3-030-44932-2_2.

[24] Valle CA, Beasley JE, Da Cunha AS. Optimally solving the joint order batching and picker routing problem. *European Journal of Operational Research*. 2017;262(3):817-834. DOI: 10.1016/j.ejor.2017.03.069.

[25] Yang N. Evaluation of the joint impact of the storage assignment and order batching in mobile-pod warehouse systems. *Mathematical Problems in Engineering*. 2022;(1):9148001. DOI: 10.1155/2022/9148001.

[26] Zhao A, Bard JF. Batch scheduling in a multi-purpose system with machine downtime and a multi-skilled workforce. *International Journal of Production Research*. 2024;62(12):4470-4493. DOI: 10.1080/00207543.2023.2265508.

[27] Xiang X, Liu C, Miao L. Storage assignment and order batching problem in Kiva mobile fulfilment system. *Engineering Optimization*. 2018;50(11):1941-1962. DOI: 10.1080/0305215X.2017.1419346.

[28] Nicolas L, Yannick F, Ramzi H. Order batching in an automated warehouse with several vertical lift modules: Optimization and experiments with real data. *European Journal of Operational Research*. 2018;267(3):958-976. DOI: 10.1016/j.ejor.2017.12.037.

[29] Jiang H. Solving multi-robot picking problem in warehouses: A simulation approach. *International Journal of Simulation Modelling*. 2020;19(4):701-712. DOI: 10.2507/ijsimm19-4-co19.

[30] Hu KY, Chang TS. An innovative automated storage and retrieval system for B2C e-commerce logistics. *The International Journal of Advanced Manufacturing Technology*. 2010;48:297-305. DOI: 10.1007/s00170-009-2292-4.

[31] Lei B, et al. Optimization of storage location assignment in tier-to-tier shuttle-based storage and retrieval systems based on mixed storage. *Mathematical Problems in Engineering*. 2020;(1):2404515. DOI: 10.1155/2020/2404515.

[32] Wang Y, et al. Modeling of parallel movement for deep-lane unit load autonomous shuttle and stacker crane warehousing systems. *Processes*. 2020;8(1):80. DOI: 10.3390/pr8010080.

[33] Winkelhaus S, et al. Hybrid order picking: A simulation model of a joint manual and autonomous order picking system. *Computers & Industrial Engineering*. 2022;167:107981. DOI: 10.1016/j.cie.2022.107981.

[34] Liang K, et al. Research on a dynamic task update assignment strategy based on a "parts to picker" picking system. *Mathematics*. 2023;11(7):1684. DOI: 10.3390/math11071684.

[35] Xie L, Li H, Luttmann L. Formulating and solving integrated order batching and routing in multi-depot AGV-assisted mixed-shelves warehouses. *European Journal of Operational Research*. 2023;307(2):713-730. DOI: 10.1016/j.ejor.2022.08.047.

[36] Kucuksari Z. *Optimal order batching for automated warehouse picking.* PhD thesis. University of Waterloo; 2023.

[37] Bansal V, Roy D. Stochastic modeling of multiline orders in integrated storage-order picking system. *Naval Research Logistics (NRL)*. 2021;68(6):810-836. DOI: 10.1002/nav.21978.

[38] Wang X, et al. Dynamic multi-tour order picking in an automotive-part warehouse based on attention-aware deep reinforcement learning. *Robotics and Computer-Integrated Manufacturing*. 2025;94:102959. DOI: 10.1016/j.rcim.2025.102959.

[39] Zhen L, et al. How to deploy robotic mobile fulfillment systems. *Transportation Science*. 2023;57(6):1671-1695. DOI: 10.1287/trsc.2022.0265.