

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ  
ТЕХНОЛОГІЙ

Кваліфікаційна наукова  
праця на правах рукопису

БАЛВАК АНДРІЙ АНАТОЛІЙОВИЧ

УДК 004.8:004.94:658.7

**ДИСЕРТАЦІЯ**  
**МОДЕЛЬ ТА МЕТОД ОПТИМІЗАЦІЇ ОБРОБКИ ІНФОРМАЦІЇ В**  
**ЛОГІСТИЧНИХ ЦЕНТРАХ НА ОСНОВІ МАШИННОГО НАВЧАННЯ**

Спеціальність 123 «Комп'ютерна інженерія»

Галузь знань 12 «Інформаційні технології»

Подається на здобуття наукового ступеня доктора філософії

Дисертація містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

\_\_\_\_\_ Андрій БАЛВАК

Науковий керівник Лащевська Наталія Олександрівна, к.т.н., доцент

Київ – 2026

## АНОТАЦІЯ

*Балвак А.А.* Модель та метод оптимізації обробки інформації в логістичних центрах на основі машинного навчання. – Кваліфікаційна наукова праця на правах рукопису.

Дисертація на здобуття наукового ступеня доктора філософії в галузі знань 12 – Інформаційні технології за спеціальністю 123 – Комп’ютерна інженерія. – Державний університет інформаційно-комунікаційних технологій. – Київ, 2026.

Дисертаційна робота присвячена актуальній науковій задачі розробки та впровадження інтелектуальних методів обробки інформації для оптимізації функціонування логістичних центрів в умовах активної інтеграції інтелектуальних інформаційних систем. Актуальність проведеного дослідження визначається стрімким переходом до парадигми «Логістика 4.0», яка вимагає перетворення складських комплексів з пасивних місць зберігання у високотехнологічні кіберфізичні системи. У подібних середовищах ефективність операцій залежить не лише від механізації праці, а й від швидкості та якості обробки великих масивів даних.

У межах дослідження проведено аналіз та запропоновано шляхи розв’язання проблеми неефективності складських процесів, зумовленої використанням негнучких алгоритмів. У роботі підкреслено обмеженість традиційних підходів в системах управління складом, які зазвичай базуються на статичних правилах ABC та XYZ-аналізу та фіксованих регламентах. Такі методи часто ігнорують часову структуру попиту, що призводить до неоптимального використання простору та надмірних витрат часу на комплектацію замовлень у динамічному ринковому середовищі. Метою дослідження є розробка нових моделей кластеризації на основі аналізу часових рядів попиту та створення методики Часово-орієнтованого Призначення Місць Зберігання (TOASL), яка дозволяє мінімізувати сумарні маршрути складських операцій.

Логіку та структуру проведеного дослідження розкрито за розділами наступним чином:

У Розділі 1 проаналізовано сучасний стан алгоритмічного та технологічного забезпечення логістики. Розглянуто роль Інтернету речей, технологій радіочастотної ідентифікації та автономних мобільних роботів у формуванні інформаційної бази складу. Обґрунтовано необхідність переходу від дискретних транзакційних записів до векторного представлення даних у формі часових рядів для збереження повної інформації про структуру споживання.

Другий розділ спрямований на дослідження математичних методів кластеризації часових рядів у логістиці для виявлення та аналізу прихованих залежностей у замовленнях товарів. Це дозволяє здійснювати перехід від статичного опису асортименту до виявлення динамічних закономірностей попиту. У межах розділу проведено порівняльний аналіз метрик відстані, таких як Euclidean, Dynamic Time Warping та Shape-Based Distance. Визначено, що застосування цих метрик дозволяє ефективно групувати товари за подібністю фазових зсувів, зумовлених часовим зміщенням піків попиту та загальної форми часових рядів. Окрему увагу приділено обґрунтуванню кількості кластерів за допомогою методу ліктя та коефіцієнта силуету.

У Розділі 3 описано розроблене симуляційне середовище для верифікації моделей розміщення. Спочатку обчислено сумарні відстані, які долаються персоналом для укладання товарів на місця зберігання. На наступному етапі порівняно дистанції, пройдені безпосередньо під час збирання товарів для замовлень різного обсягу. Оцінювання здійснювалося на основі п'яти політик маршрутизації, зокрема S-shape, mid-point, largest gap, return та composite за умов реалізації стратегій розміщення товарів Random, ABC, COL і TOASL. Це дозволило визначити ефективність досліджуваних підходів у контексті мінімізації загальних маршрутів комплектування. Статистичну значущість результатів підтверджено за допомогою методів ANOVA та критерію Тьюкі.

Наукові результати, отримані в дисертаційній роботі:

1. Вперше розроблено метод Time-Oriented Assignment of Storage Locations (TOASL) для інтелектуального розміщення товарів у логістичних центрах, який за рахунок використання кластеризації часових рядів параметрів попиту з використанням метрик DTW та SBD, урахування динамічних характеристик попиту (сезонність, тренди, варіативність), адаптивного зонування складського простору залежно від кластерів попиту, дозволило сформувати відображення «товар–зона» як функцію поведінки попиту в часі та забезпечити зменшення середньої довжини маршрутів комплектування без зміни складської інфраструктури.

2. Удосконалено модель оцінювання ефективності складських операцій, яка за рахунок використання багатокритеріальної оптимізацію шляхом використання комбінованої цільової функції, що інтегрує довжину маршрутів комплектування, витрати на розміщення товарів та часові характеристики виконання операцій, що дозволило досягти зниження сумарних операційних витрат на розміщення та обробку замовлень.

3. Удосконалено модель інтелектуального управління складськими процесами, яка за рахунок використання математичного переходу до векторного представлення попиту у вигляді часових профілів, застосування ієрархічної кластеризації та самоорганізовуваних карт Кохонена, використання спеціалізованих метрик подібності часових рядів, дозволило підвищити точність кластеризації товарів за поведінкою показників попиту, забезпечити зменшення сумарної довжини маршрутів і підвищення швидкості виконання складських операцій та сформувати цілісну інтегровану модель управління складом, орієнтовану на мінімізацію логістичних витрат й підвищення продуктивності без розширення інфраструктури.

Авторський внесок полягає в розробці алгоритмічного та програмного забезпечення для обробки транзакційних даних і формуванні методики побудови нормованих часових рядів. Також запропоновано гібридну політику розміщення, що базується на групуванні та спільному розташуванні товарів, які часто замовляються разом, у поєднанні з локальним ABC-впорядкуванням.

Отримані наукові результати показали, що застосування методики TOASL за комбінованої стратегії маршрутизації дозволяє досягти скорочення середньої довжини маршрутів комплектування на 32% відносно випадкового розміщення та на 15% порівняно з ABC. Експериментально доведено забезпечення TOASL стабільно коротших маршрутів збирання за різних розмірів партій замовлень. Водночас встановлено, що при впровадженні TOASL витрати на первинне розміщення товарів зростають лише на 5,3% порівняно зі стратегією найближчої вільної комірки, що нівелюється вагогим підвищенням ефективності операцій на етапі відбору замовлень.

Практичне значення отриманих результатів полягає у доведенні до рівня практичного застосування методики інтелектуального розміщення товарів у логістичних центрах на основі аналізу часових рядів попиту. Запропоновані в дисертації моделі та алгоритми можуть бути використані в діяльності логістичних центрів, складських комплексів, e-commerce-платформ та підприємств дистрибуції для підвищення ефективності управління складськими операціями. Експериментально встановлено, що впровадження методики TOASL дає змогу скоротити середню довжину маршрутів комплектування на 32% відносно випадкового розміщення та на 15% порівняно з традиційною ABC-стратегією, зберігаючи перевагу для різних розмірів партій замовлень. Водночас приріст витрат на первинне розміщення товарів не перевищує 5,3%, що свідчить про збалансованість запропонованого підходу та його придатність до практичного використання. Реалізація отриманих результатів у WMS-системах дозволяє знизити операційні витрати, зменшити інтенсивність переміщень у складському просторі, підвищити швидкість виконання замовлень і покращити якість логістичного сервісу без необхідності розширення фізичної інфраструктури складу. Дисертаційна робота була виконана в рамках госпдоговірної науково-дослідної роботи «Комплексна розробка прикладних ІТ-рішень для підвищення продуктивності комп'ютерних систем у комерційному та соціальному секторі» (Державний

реєстраційний номер 0125U003178), Державного університету інформаційно-комунікаційних технологій.

**Ключові слова:** інформаційні технології, комп'ютерні системи, оптимізація процесів, логістичні системи, обробка даних, часові ряди, кластеризація даних, машинне навчання, інтелектуальні системи, моделювання, системи управління, складська логістика, інформаційні системи, аналіз даних, оптимізаційні моделі.

## ABSTRACT

*Balvak A.A. Model and Method for Optimization of Information Processing in Logistics Centers Based on Machine Learning. – Qualification scientific work as a manuscript.*

The dissertation for the degree of Doctor of Philosophy in the field of knowledge 12 – Information Technologies, specialty 123 – Computer Engineering. – State University of Information and Communication Technologies. – Kyiv, 2026.

The dissertation is devoted to a current scientific problem of developing and implementing intelligent information processing methods for optimizing the functioning of logistics centers under conditions of active integration of intelligent information systems. The relevance of the research is determined by the rapid transition to the “Logistics 4.0” paradigm, which requires the transformation of warehouse complexes from passive storage facilities into high-tech cyber-physical systems. In such environments, operational efficiency depends not only on labor mechanization but also on the speed and quality of processing large volumes of data.

Within the study, an analysis was conducted, and approaches to solving the problem of inefficiency in warehouse processes caused by the use of inflexible algorithms were proposed. The work emphasizes the limitations of traditional approaches in warehouse management systems, which are usually based on static rules of ABC and XYZ analysis and fixed regulations. Such methods often ignore the temporal structure of demand, leading to suboptimal space utilization and excessive time consumption for order picking in a dynamic market environment. The aim of the research is to develop new clustering models based on demand time series analysis and to create a Time-Oriented Assignment of Storage Locations (TOASL) methodology that minimizes the total routes of warehouse operations.

The logic and structure of the research are presented as follows:

In Chapter 1, the current state of algorithmic and technological support in logistics is analyzed. The role of the Internet of Things, radio frequency identification technologies, and autonomous mobile robots in forming the

warehouse information base is considered. The necessity of transitioning from discrete transactional records to vector data representation in the form of time series to preserve complete information about consumption patterns is substantiated.

Chapter 2 focuses on the study of mathematical methods for clustering time series in logistics to identify and analyze hidden dependencies in product orders. This enables the transition from a static description of the assortment to the identification of dynamic demand patterns. A comparative analysis of distance metrics such as Euclidean, Dynamic Time Warping, and Shape-Based Distance is carried out. It is determined that these metrics allow efficient grouping of products based on similarity in phase shifts caused by the temporal displacement of demand peaks and overall time series shape. Particular attention is paid to determining the optimal number of clusters using the elbow method and the silhouette coefficient.

In Chapter 3, a developed simulation environment for verifying storage allocation models is described. Initially, the total distances traveled by personnel for placing goods into storage locations are calculated. At the next stage, distances traveled during order picking for orders of different sizes are compared. The evaluation is performed based on five routing policies, namely S-shape, mid-point, largest gap, return, and composite, under implementation of storage strategies Random, ABC, COL, and TOASL. This allowed determining the effectiveness of the studied approaches in minimizing total picking routes. The statistical significance of the results is confirmed using ANOVA and Tukey's test.

The scientific results obtained in the dissertation are as follows:

1. For the first time, a Time-Oriented Assignment of Storage Locations (TOASL) method for intelligent product placement in logistics centers has been developed. The method is based on clustering demand time series using DTW and SBD metrics, taking into account dynamic demand characteristics (seasonality, trends, variability), and adaptive zoning of warehouse space depending on demand clusters. This approach enables the formation of a "product-zone" mapping as a function of demand behavior over time and ensures a reduction in the average picking route length without changing the warehouse infrastructure.



2. The model for evaluating the efficiency of warehouse operations has been improved through the application of multi-criteria optimization using a combined objective function that integrates picking route length, product placement costs, and time characteristics of operations. This approach enables a reduction in total operational costs associated with product placement and order processing.

3. The model of intelligent warehouse process management has been improved by introducing a mathematical transition to vector representation of demand in the form of temporal profiles, applying hierarchical clustering and Kohonen self-organizing maps, and using specialized similarity metrics for time series. This allows increasing the accuracy of product clustering based on demand behavior, reducing total route length, improving the speed of warehouse operations, and forming an integrated warehouse management model aimed at minimizing logistics costs and increasing productivity without expanding infrastructure.

The author's contribution consists in the development of algorithmic and software tools for processing transactional data and constructing normalized time series. A hybrid storage policy is also proposed, based on grouping and co-location of frequently co-ordered products combined with local ABC ordering.

The obtained scientific results demonstrate that the application of the TOASL methodology with a combined routing strategy allows reducing the average order picking route length by 32% compared to random placement and by 15% compared to the ABC strategy. It is experimentally proven that TOASL ensures consistently shorter picking routes for different order sizes. At the same time, it is established that the cost of initial product placement increases by only 5.3% compared to the nearest available location strategy, which is compensated by a significant increase in efficiency at the order picking stage.

The practical significance of the results lies in bringing the developed methodology of intelligent product placement in logistics centers to the level of practical application based on demand time series analysis. The proposed models and algorithms can be used in logistics centers, warehouse complexes, e-commerce platforms, and distribution enterprises to improve warehouse operations

management. It has been experimentally established that the implementation of the TOASL methodology reduces the average picking route length by 32% compared to random placement and by 15% compared to the traditional ABC strategy, while maintaining efficiency across different order sizes. At the same time, the increase in initial placement costs does not exceed 5.3%, confirming the balance and practical applicability of the approach. Implementation in WMS systems allows reducing operational costs, decreasing unnecessary movements within warehouse space, increasing order processing speed, and improving logistics service quality without expanding physical infrastructure.

The dissertation was carried out within the framework of the contractual research project “Comprehensive Development of Applied IT Solutions for Improving the Performance of Computer Systems in the Commercial and Social Sectors” (State Registration No. 0125U003178), State University of Information and Communication Technologies.

**Keywords:** information technologies, computer systems, process optimization, logistics systems, data processing, time series, data clustering, machine learning, intelligent systems, modeling, management systems, warehouse logistics, information systems, data analysis, optimization models.

## Список опублікованих праць за темою дисертації

### *Матеріали й тези наукових конференцій*

1. Лемешко А., Балвак А. АНАЛІЗ ЕФЕКТИВНОСТІ АЛГОРИТМІВ КЛАСТЕРИЗАЦІЇ ДЛЯ ОПТИМІЗАЦІЇ РОЗТАШУВАННЯ ТОВАРІВ НА СКЛАДАХ. *Наукова конференція молодих вчених «Інновації»* : Зб. тез, м. Київ, 19 верес. 2024 р. Київ, 2024. С. 23–24
2. Балвак А., Лащевська Н. САМООРГАНІЗУЮЧІ КАРТИ ЯК ІНСТРУМЕНТ ДЛЯ РАЦІОНАЛЬНОГО РОЗТАШУВАННЯ ТОВАРІВ НА СКЛАДІ. *Науково-практична конференція «Проблеми комп'ютерної інженерії»* : Зб. тез, м. Київ, 3 груд. 2024 р. Київ, 2024. С. 66–68
3. Балвак А., Лащевська Н. РОЛЬ ІНТЕРНЕТУ РЕЧЕЙ В УПРАВЛІННІ СКЛАДАМИ. *СУЧАСНИЙ СТАН ТА ПЕРСПЕКТИВИ РОЗВИТКУ ІОТ* : Зб. тез, м. Київ, 15 квіт. 2025 р. Київ, 2025. С. 210–212
4. Балвак А., Лащевська Н. Інтеграція технологій машинного навчання та штучного інтелекту в системи управління складом. *Застосування програмного забезпечення в інформаційно-комунікаційних технологіях* : Зб. тез, м. Київ, 24 квіт. 2025 р. Київ, 2025. С. 308–310
5. Балвак А., Лащевська Н. ПРОГРАМНА МЕТОДИКА ОПТИМІЗАЦІЇ РОЗМІЩЕННЯ ТОВАРІВ НА ОСНОВІ АНАЛІЗУ ЧАСОВИХ РЯДІВ ПОПИТУ. *VI науково-практична конференція «Проблеми комп'ютерної інженерії»* : Зб. тез, м. Київ, 3 груд. 2025 р. Київ, 2025. С. 47–49

### *Статті в наукових фахових виданнях*

1. Лемешко А. В., Балвак А. ОПТИМІЗАЦІЯ ОБРОБЛЕННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ СУЧАСНИХ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У РОБОТІ ЛОГІСТИЧНИХ ЦЕНТРІВ. *ЗВ'ЯЗОК*. 2022. № 2. С. 66–69. URL: <https://doi.org/10.31673/2412-9070.2022.026669>
2. АКТУАЛЬНІ ЗАСАДИ СТВОРЕННЯ АЛГОРИТМІВ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ЛОГІСТИЧНИХ ЦЕНТРІВ / А. В. Лемешко та

ін. *Таврійський науковий вісник. Серія: Технічні науки*. 2023. № 1. С. 25–32. URL: <https://doi.org/10.32851/tnv-tech.2023.1.3>

3. ОБРОБКА ТА АНАЛІЗ ДАНИХ НА ПРИКЛАДІ НАБОРУ SPAMBASE З ВИКОРИСТАННЯМ БІБЛІОТЕК ДЛЯ МАШИННОГО НАВЧАННЯ / А. А. Балвак та ін. *Таврійський науковий вісник. Серія: Технічні науки*. 2024. № 2. С. 3–20. URL: <https://doi.org/10.32782/tnv-tech.2024.2.1>

4. Балвак А., Лащевська Н. ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО ОПТИМІЗАЦІЇ СКЛАДСЬКИХ ПРОЦЕСІВ У ЗАДАЧАХ РОЗМІЩЕННЯ КОМПЛЕКТУВАННЯ ТА РОБОТИЗАЦІЇ. *Кібербезпека: освіта, наука, техніка*. 2025. Т. 1, № 29. С. 161–177. URL: <https://doi.org/10.28925/2663-4023.2025.29.869>

5. Балвак А., Лащевська Н. МЕТОДИКА ЗБАЛАНСОВАНОГО ПРИЗНАЧЕННЯ МІСЦЬ ЗБЕРІГАННЯ ТОВАРІВ НА ОСНОВІ ЧАСОВИХ ХАРАКТЕРИСТИК ПОПИТУ. *Кібербезпека: освіта, наука, техніка*. 2025 Т. 3, № 31. С. 155–168. URL: <https://doi.org/10.28925/2663-4023.2025.31.1002>

6. Балвак А.А. Зінченко В.В. ФОРМУВАННЯ ВИБІРКИ ДАНИХ ТА КРИТЕРІЇ ЯКОСТІ ДЛЯ МОДЕЛЮВАННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ СКЛАДСЬКОЇ ЛОГІСТИКИ. *ЗВ'ЯЗОК*. 2026. № 1. С. 96-104. <https://doi.org/10.31673/2412-9070.2026.017401>

*Статті в наукових фахових виданнях, що індексуються в міжнародних  
базах Scopus*

1. Integrated Modeling in the Quality Assessment of Flight Management Software Systems / О. Zinchenko та ін. *Informatica*. 2025. Т. 49, № 31. URL: <https://doi.org/10.31449/inf.v49i31.11012>

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ .....	15
ВСТУП .....	17
РОЗДІЛ 1. АЛГОРИТМІЧНЕ ТА ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ СКЛАДСЬКОЇ ЛОГІСТИКИ.....	23
1.1 Концептуальні засади групування товарів та методи оптимізації маршрутів .....	24
1.2 Адаптація розміщення товарів за допомогою машинного навчання .....	31
1.3 Автоматизація збору товарів, управління запасами та роль IoT .....	35
1.4 Методологічні основи аналізу даних, часові ряди та формування вибірок .....	43
1.5 Досвід провідних компаній у впровадженні сучасних технологій у складській логістиці .....	48
Висновки до розділу 1 .....	52
РОЗДІЛ 2. МЕТОДИ КЛАСТЕРИЗАЦІЇ ТОВАРІВ НА ОСНОВІ ЧАСОВИХ РЯДІВ ....	53
2.1 Побудова часових рядів попиту на товари .....	53
2.2. Алгоритми кластеризації часових рядів.....	62
2.3. Вибір оптимального кластерного розбиття для цілей зберігання.....	77
Висновки до розділу 2 .....	85
РОЗДІЛ 3. РОЗРОБКА ТА ПЕРЕВІРКА МОДЕЛІ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ТОВАРІВ НА СКЛАДІ .....	86
3.1 Формалізація критерію оптимізації з урахуванням довжин маршрутів.....	86
3.2 Побудова симуляційного середовища для оцінки розміщення .....	94
3.3 Опис вихідних даних та їх підготовка .....	97
3.4 Кластеризація часових рядів та вибір кількості кластерів .....	104
3.5 Конфігурація складу і параметри симуляції.....	115
3.6 Моделювання та порівняння стратегій розміщення товарів на складі.....	118
3.7 Стратегії комплектування замовлень на складі .....	127
3.8 Теоретичні засади Box Plot, однофакторної ANOVA та критерію Тьюкі HSD .....	133
3.9 Експериментальні результати та їх статистична оцінка .....	138
Висновки до розділу 3 .....	152
ВИСНОВКИ.....	154
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	156
Додаток А – Лістинг коду алгоритму підготовки даних та виділення Pareto-групи	169
Додаток Б – Лістинг коду формування ознак, PCA та агломеративної кластеризації товарів .....	171

Додаток В – Лістинг коду побудови часових рядів і кластеризації з визначенням оптимального числа кластерів.....	174
Додаток Г – Лістинг коду ANOVA для групових порівнянь і пост-хок аналізу Tukey HSD.....	175

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

ABC – метод класифікації запасів за ступенем їхнього внеску в загальний результат (ABC-аналіз)

АНС – агломеративна ієрархічна кластеризація (Agglomerative Hierarchical Clustering)

AMR – автономні мобільні роботи (Autonomous Mobile Robots)

ANOVA – дисперсійний аналіз (ANalysis Of VAriance)

COL – стратегія розміщення товарів у найближчій вільній комірці (Closest Open Location)

CPS – кіберфізична система (Cyber-Physical System)

DTW – алгоритм динамічної трансформації часової шкали (Dynamic Time Warping)

ERP – система планування ресурсів підприємства (Enterprise Resource Planning)

HSD – критерій значущої різниці Тьюкі (Honestly Significant Difference)

IoT – Інтернет речей (Internet of Things)

KPI – ключові показники ефективності (Key Performance Indicators)

ML – машинне навчання (Machine Learning)

RFID – радіочастотна ідентифікація (Radio Frequency IDentification)

SBD – міра відстані на основі форми часових рядів (Shape-Based Distance)

SKU – одиниця складського обліку (Stock Keeping Unit)

SLAP – задача призначення місць зберігання (Storage Location Assignment Problem)

SOM – самоорганізаційна карта (Self-Organizing Map)

TOASL – методика Часово-орієнтованого Призначення Місць Зберігання (Time-Oriented Assignment of Storage Locations)

TSP – задача комівояжера (Travelling Salesman Problem)

VEN – метод класифікації запасів за їхньою критичністю для процесів (Vital, Essential, Non-essential)

VRP – задача маршрутизації транспортних засобів (Vehicle Routing Problem)

WCSS – внутрішньокластерна сума квадратів (Within-Cluster Sum of Squares)

WMS – система управління складом (Warehouse Management System)

XYZ – метод класифікації товарів за стабільністю та прогнозованістю попиту



## ВСТУП

**Обґрунтування вибору теми дослідження.** Сучасний етап розвитку глобальної економіки характеризується стрімкою цифровізацією та фундаментальною трансформацією ланцюгів постачання. Логістичні центри, які раніше розглядалися переважно як пасивні зони зберігання запасів, сьогодні перетворюються на ключові вузли інтелектуальних мереж розподілу. Актуальність проведеного дослідження визначається переходом до парадигми «Логістика 4.0», що передбачає створення високотехнологічних кіберфізичних систем. У таких середовищах фізичні операції переміщення товарів тісно інтегровані з потужними обчислювальними алгоритмами, а загальна ефективність бізнес-процесів залежить не лише від механізації, а й від швидкості та якості обробки великих масивів даних.

Стрімке зростання сектору електронної комерції та поширення омніканальних моделей торгівлі висувають нові вимоги до систем управління складом (WMS). Головна науково-практична проблема полягає в обмеженості традиційних підходів до організації складських процесів. Більшість існуючих WMS базуються на статичних правилах розміщення, таких як класичний аналіз ABC, XYZ. Хоча ці методи є простими в реалізації, вони оперують агрегованими показниками за тривалі періоди, що призводить до ігнорування часової структури попиту. В умовах динамічного ринку це спричиняє ситуації за яких для товарів з кардинально різними патернами споживання, зокрема сезонної продукції та позицій зі стабільним попитом, застосовуються ідентичні стратегії зберігання, що призводить до неоптимального використання простору та значних часових витрат на комплектування замовлень.

Вагомий внесок у дослідження процесів обробки інформації та автоматизації складської логістики зробили вчені de Koster R., Roodbergen K. J., Azizpour M., Namazypour N., Pang K.-W., Chan H.-L., Allgor R., Cezik T., Chen H., Wang Z [8-12]. Попри значні результати, у працях

цих науковців залишаються недостатньо вирішеними питання динамічної адаптації стратегій розміщення до змінних часових характеристик попиту та групування асортименту на основі схожості патернів споживання. Відсутність цілісних підходів до інтелектуального аналізу часових рядів у задачах оптимізації топології складу визначає актуальність подальших розробок у цьому напрямі.

Таким чином, постає об'єктивна необхідність у розробці нових інтелектуальних методів обробки інформації, які б забезпечили перехід від статичного зонування до динамічного управління на основі аналізу часових рядів. Використання технологій Інтернету речей, радіочастотної ідентифікації та автономних мобільних роботів створює необхідну високодеталізовану базу даних, що дозволяє застосовувати сучасні алгоритми машинного навчання для оптимізації роботи логістичних центрів.

**Зв'язок роботи з науковими програмами, планами, темами.** Дисертаційна робота була виконана в рамках госпдоговірної науково-дослідної роботи «Комплексна розробка прикладних IT-рішень для підвищення продуктивності комп'ютерних систем у комерційному та соціальному секторі» (Державний реєстраційний номер 0125U003178), Державного університету інформаційно-комунікаційних технологій.

**Мета і завдання дослідження.** Мета дослідження полягає у розробці та науковому обґрунтуванні моделей кластеризації попиту на основі аналізу часових рядів, а також створенні методу оптимізації складських операцій, спрямованої на мінімізацію сумарної довжини маршрутів переміщення з використанням інтелектуальних методів обробки даних.

Для досягнення поставленої мети було визначено та розв'язано наступні задачі:

1. Проаналізувати сучасний стан алгоритмічного та технологічного забезпечення складської логістики, дослідити роль Індустрії 4.0 та Інтернету речей у формуванні інформаційної бази складу.

2. Обґрунтувати необхідність переходу від дискретних транзакційних записів до векторного представлення даних у формі часових рядів для збереження повної інформації про структуру споживання.

3. Дослідити математичні методи кластеризації часових рядів та провести порівняльний аналіз метрик відстані для виявлення прихованих залежностей у попиті.

4. Розробити методику Часово-орієнтованого Призначення Місць Зберігання (TOASL), що інтегрує результати кластерного аналізу з динамічним зонуванням складу та внутрішньозоновою оптимізацією.

5. Формалізувати комбінований критерій оптимізації, який одночасно враховує витрати на розміщення товарів на зберігання та їх подальше комплектування.

6. Створити симуляційне середовище для верифікації розроблених моделей на основі реальних транзакційних даних інтернет-ритейлера.

7. Провести багатофакторну статистичну оцінку ефективності запропонованої методики TOASL порівняно з традиційними стратегіями розміщення товарів на зберігання (Random, ABC, COL) за різних політик маршрутизації комплектувальників.

**Об'єктом дослідження** є процеси обробки інформації для організації зберігання та відбору товарів у логістичних центрах.

**Предметом дослідження** є моделі, алгоритми та методи інтелектуальної оптимізації інформаційних потоків і розміщення ресурсів у логістичних центрах на основі аналізу часових рядів.

**Методами дослідження** є системний аналіз, методи машинного навчання, математичне моделювання, симуляційне моделювання та методи математичної статистики.

**Наукова новизна отриманих результатів** полягає у наступному:

1. Вперше розроблено метод Time-Oriented Assignment of Storage Locations (TOASL) для інтелектуального розміщення товарів у логістичних центрах, який за рахунок використання кластеризації часових рядів параметрів

попиту з використанням метрик DTW та SBD, урахування динамічних характеристик попиту (сезонність, тренди, варіативність), адаптивного зонування складського простору залежно від кластерів попиту, дозволило сформувати відображення «товар–зона» як функцію поведінки попиту в часі та забезпечити зменшення середньої довжини маршрутів комплектування без зміни складської інфраструктури.

2. Удосконалено модель оцінювання ефективності складських операцій, яка за рахунок використання багатокритеріальної оптимізацію шляхом використання комбінованої цільової функції, що інтегрує довжину маршрутів комплектування, витрати на розміщення товарів та часові характеристики виконання операцій, що дозволило досягти зниження сумарних операційних витрат на розміщення та обробку замовлень.

3. Удосконалено модель інтелектуального управління складськими процесами, яка за рахунок використання математичного переходу до векторного представлення попиту у вигляді часових профілів, застосування ієрархічної кластеризації та самоорганізовуваних карт Кохонена, використання спеціалізованих метрик подібності часових рядів, дозволило підвищити точність кластеризації товарів за поведінкою показників попиту, забезпечити зменшення сумарної довжини маршрутів і підвищення швидкості виконання складських операцій та сформувати цілісну інтегровану модель управління складом, орієнтовану на мінімізацію логістичних витрат й підвищення продуктивності без розширення інфраструктури.

**Практичне значення отриманих результатів** полягає у доведенні до рівня практичного застосування методики інтелектуального розміщення товарів у логістичних центрах на основі аналізу часових рядів попиту. Запропоновані в дисертації моделі та алгоритми можуть бути використані в діяльності логістичних центрів, складських комплексів, e-commerce-платформ та підприємств дистрибуції для підвищення ефективності управління складськими операціями. Експериментально встановлено, що впровадження методики TOASL дає змогу скоротити середню довжину маршрутів

комплектування на 32% відносно випадкового розміщення та на 15% порівняно з традиційною ABC-стратегією, зберігаючи перевагу для різних розмірів партій замовлень. Водночас приріст витрат на первинне розміщення товарів не перевищує 5,3 %, що свідчить про збалансованість запропонованого підходу та його придатність до практичного використання. Реалізація отриманих результатів у WMS-системах дозволяє знизити операційні витрати, зменшити інтенсивність переміщень у складському просторі, підвищити швидкість виконання замовлень і покращити якість логістичного сервісу без необхідності розширення фізичної інфраструктури складу.

**Особистий внесок здобувача.** Дисертація є самостійним науковим дослідженням. Всі результати, що виносяться на захист, отримані автором особисто. Здобувачем розроблено алгоритмічне та програмне забезпечення для обробки великих масивів транзакційних даних, сформовано методику побудови нормованих часових рядів та запропоновано гібридну політику розміщення.

**Апробація матеріалів дисертації.** Основні положення та наукові результати дисертаційної роботи доповідалися та обговорювалися на науково-практичних конференціях Всеукраїнського рівня:

- Наукова конференція молодих вчених «Інновації» (Київ, 19 вересня 2024 р.).
- Науково-практична конференція «Проблеми комп'ютерної інженерії» (Київ, 3 грудня 2024 р.).
- VI Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку IoT» (Київ, 15 квітня 2025 р.).
- VI Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інформаційно-комунікаційних технологіях». (Київ, 24 квітня 2025 р.).
- VI науково-практична конференція «Проблеми комп'ютерної інженерії» (Київ, 3 грудня 2025 р.).

Основні наукові положення та результати дисертаційного дослідження опубліковано в 3 наукових працях у періодичних виданнях України включених до Переліку наукових фахових видань України.

**Структура та обсяг дисертації.** Робота складається з анотації, змісту, переліку умовних позначень, вступу, трьох розділів, загальних висновків, списку використаних джерел (105 найменувань) та 4 додатків. Загальний обсяг дисертації становить 175 сторінок машинописного тексту, включаючи 17 таблиць та 22 рисунки.

.

## РОЗДІЛ 1. АЛГОРИТМІЧНЕ ТА ТЕХНОЛОГІЧНЕ ЗАБЕЗПЕЧЕННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ СКЛАДСЬКОЇ ЛОГІСТИКИ

Сучасні логістичні центри є ключовими вузлами глобальних ланцюгів постачання, забезпечуючи комплексні функції зберігання, консолідації та розподілу товарних потоків. Проте на тлі стрімкої цифровізації економіки відбувається перехід до концепцій Індустрії 4.0, яка знаменує новий етап промислового розвитку та характеризується масовим упровадженням інтелектуальних цифрових технологій та автоматизації у виробничі й логістичні процеси. Внаслідок цих глобальних зрушень роль складських комплексів зазнає фундаментальних змін. Вони трансформуються з пасивних місць зберігання запасів у високотехнологічні автоматизовані комплекси. У них фізичні процеси переміщення товарів тісно інтегровані з потужними обчислювальними алгоритмами, а ефективність операцій залежить не лише від механізації праці, а й від швидкості та якості обробки інформації [7].

Актуальність цифровізації логістики зумовлена необхідністю адаптації до нових ринкових викликів, які виникли внаслідок стрімкого зростання сфери електронної комерції та поширення омніканальної торгівлі. Остання є моделлю продажів, яка об'єднує різні канали комунікації, такі як онлайн-магазини, мобільні додатки та фізичні точки в єдину інтегровану систему обслуговування клієнта. Згідно з дослідженням [13], парадигма «**Логістика 4.0**» (*Logistics 4.0*) передбачає не просто автоматизацію окремих процесів, а створення взаємопов'язаних систем, де люди, інтелектуальні машини та об'єкти (товари) взаємодіють у реальному часі для підвищення гнучкості та ефективності.

Основною науково-практичною проблемою є обмеженість традиційних підходів до управління складом. Існуючі **системи управління складом** (*Warehouse Management System, WMS*) часто базуються на статичних правилах розміщення та фіксованих алгоритмах, які були розроблені для стабільних умов попиту. Проте, як вказано в роботі [14], стандартні операційні процеси,

хоч і важливі для контролю, часто унеможливають динамічне реагування на обмеження реального часу та щоденні операційні збої, характерні для середовища з динамічним попитом. Статичні методи управління виявляються нездатними ефективно реагувати на непередбачувані коливання обсягів замовлень, що призводить до неоптимального використання складського простору та надмірних витрат часу на комплектацію.

Відтак, спостерігається необхідність переходу від статичних, детермінованих моделей управління до динамічних та інтелектуальних систем. Цей перехід стає можливим завдяки інтеграції технологій **Інтернету речей** (*Internet of Things, IoT*), що забезпечують безперервний моніторинг параметрів і наскрізну видимість товарних потоків для формування **великих масивів даних** (*Big Data*) із засобами штучного інтелекту та **машинного навчання** (*Machine Learning, ML*). Останні дозволяють виявляти приховані закономірності в історії операцій, прогнозувати попит та адаптивно оптимізувати маршрутизацію і розміщення товарів, забезпечуючи гнучкість прийняття рішень у реальному часі.

В цьому розділі буде проаналізовано існуючі технологічні рішення та методи обробки даних. Такий аналіз формує базу для розробки нових моделей оптимізації роботи логістичних центрів, які представлені у наступних розділах дисертації.

## **1.1 Концептуальні засади групування товарів та методи оптимізації маршрутів**

Ефективне функціонування сучасного логістичного центру базується на вирішенні взаємопов'язаних завдань раціонального розміщення товарних запасів та мінімізації витрат на їх переміщення [2]. Історичний розвиток методів управління складом демонструє еволюцію від простих детермінованих моделей класифікації до складних адаптивних алгоритмів, що використовують методи комбінаторної оптимізації та штучного інтелекту [13].



**Традиційні підходи до класифікації запасів.** Фундаментальним принципом управління складом є диференціація запасів за ступенем їхньої значущості для підприємства, що сприяє раціональному розподілу ресурсів на їх обробку та зберігання. Серед класичних методик, що інтегровані в більшість сучасних WMS та базуються на статистичному аналізі історичних даних, домінуючу роль відіграє ABC-аналіз. Цей метод структурного аналізу запасів ґрунтується на емпіричному принципі **Парето** (*Pareto*) відомому як правило 20/80 і передбачає сегментацію всієї номенклатури на три ієрархічні категорії [15]. До групи А відносять найбільш цінні товари, які складають близько 20 % асортименту, але забезпечують до 80 % загального обігу; ці позиції вимагають найсуворішого обліку та пріоритетного розміщення в зонах найкращої доступності. Група В є проміжною категорією, що охоплює близько 30 % товарних позицій та генерує 15 % обігу, тоді як група С об'єднує найменш цінні товари (50 % асортименту), внесок яких у загальний результат становить лише 5 %.

Математично процес визначення меж груп базується на розрахунку кумулятивної (накопиченої) частки  $W_i$  для кожної  $i$ -ї позиції у списку номенклатури:

$$W_i = \frac{\sum_{j=1}^i V_j}{V_{total}} \cdot 100\%, \quad (1.1)$$

де  $V_j$  – обсяг реалізації  $j$ -го найменування;  $i$  – ранг позиції у списку, відсортованому за спаданням  $V_j$ ;  $V_{total}$  – загальний обсяг продажу всіх товарів.

Для врахування нерівномірності попиту застосовується XYZ-аналіз, що передбачає класифікацію товарів за коефіцієнтом варіації попиту  $v$ :

$$v = \frac{\sigma}{\bar{x}} \cdot 100\%, \quad (1.2)$$

де  $\sigma$  – стандартне відхилення, а  $\bar{x}$  – середнє значення попиту за період. Виходячи з цього виділяють три типи товарів. До категорії X належать позиції зі стабільним споживанням для яких  $0 \leq \nu \leq 10\%$ . Клас Y включає товари з сезонними коливаннями, де  $10\% < \nu \leq 25\%$ , а сегмент Z – артикули з непередбачуваним попитом для яких  $\nu > 25\%$  [16].

Додатково для оцінки критичності запасів використовується **VEN-аналіз** (*Vital, Essential, Non-essential*), що дозволяє надавати пріоритети товарам не за фінансовими показниками, а за їхньою важливістю для безперервності технологічного процесу або виконання зобов'язань перед клієнтами [17]. Згідно з цією методикою перша категорія V або Vital позначає життєво необхідні запаси відсутність яких призводить до зупинки виробництва або суттєвих фінансових втрат, зокрема основна сировина, запчастини до унікального обладнання чи ліки першої необхідності. До другої групи E або Essential належать важливі позиції дефіцит яких є небажаним, але може бути тимчасово допущений без повної зупинки процесів, наприклад, пакувальні матеріали та стандартизовані деталі з наявними аналогами. Третя категорія N або Non-essential охоплює другорядні ресурси відсутність яких не є критичною для поточної діяльності, як-от допоміжні матеріали, інвентар для прибирання та декоративні елементи.

Систематизований огляд розглянутих вище інструментів із виокремленням їхніх ключових критеріїв, переваг та обмежень у контексті сучасних логістичних викликів представлено в табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика традиційних методів класифікації запасів

Метод аналізу	Критерій класифікації	Основна перевага	Недоліки в контексті динамічної логістики
ABC	Внесок у результат (обіг, прибуток)	Простота реалізації, фокусування на ключових ресурсах	Статичність, ігнорування коливань попиту, усереднення даних за довгий період та нівелювання піків і спадів споживання
XYZ	Стабільність (прогнозованість) споживання	Дозволяє оцінити ризик та точність прогнозу	Чутливість до аномалій, тренд не розрізняється від випадкової варіації
VEN	Критичність наявності	Забезпечення надійності системи	Суб'єктивність оцінки, складність формалізації для автоматизації через важкість переведення якісних критеріїв у чіткі алгоритми

**Критичний аналіз та обмеження.** Попри широке застосування поєднання методів ABC та XYZ має суттєвий недолік, який стає особливо відчутним в епоху Індустрії 4.0 і полягає у статичності та ігноруванні часової структури попиту. Традиційний ABC-аналіз оперує агрегованими даними за тривалий період, який може становити один рік. Такий підхід призводить до того, що позиції з яскраво вираженою сезонністю, зокрема новорічний декор та асортимент рівномірного попиту, наприклад, офісний папір, можуть потрапити до однієї групи «А» через однаковий річний обсяг продажів. Проте з погляду логістики вони вимагають кардинально різних стратегій розміщення, оскільки продукція стабільного попиту має перебувати в зоні активного відбору постійно, тоді як сезонні артикули повинні переміщуватися туди динамічно лише у піковий період. Ігнорування часової динаміки призводить до неефективного використання складського простору та подовження маршрутів комплектування у міжсезоння. Це обґрунтовує необхідність переходу від скалярних оцінок (сума продажів) до векторних

(часові ряди), що дозволить застосовувати методи кластеризації для виявлення прихованих патернів попиту [18].

**Математична формалізація логістичних задач маршрутизації.** Другою вагомою складовою ефективності складу після правильного розміщення (класифікації) є оптимізація переміщення комплектувальників. З точки зору математичного моделювання процес збору замовлень на складі зводиться до класу задач комбінаторної оптимізації. Базовою моделлю для опису маршруту одного комплектувальника є **Задача комівояжера** (*Travelling Salesman Problem, TSP*). У класичній постановці задано множину точок (місць зберігання товарів) та матрицю відстаней між ними, де необхідно знайти замкнений маршрут мінімальної довжини, що проходить через кожну точку рівно один раз.

Формально задачу можна описати на графі  $G = (V, E)$ , де  $V = \{0, 1, \dots, n\}$  – множина вершин (0 – точка старту,  $1..n$  – місця відбору), а  $E$  – множина ребер з вагами  $c_{ij}$  (відстань або час переміщення). Цільова функція мінімізації витрат виглядає наступним чином:

$$\sum_{i=0}^n \sum_{j=0, j \neq i}^n c_{ij} x_{ij} \rightarrow \min, \quad (1.3)$$

де  $x_{ij}$  – бінарна змінна, що дорівнює 1, якщо перехід від  $i$  до  $j$  включено до маршруту, і 0 в іншому випадку. Обмеження задачі повинні гарантувати відвідування кожної вершини та відсутність розривів у маршруті. Графічну інтерпретацію цієї моделі, де вершини відповідають місцям відбору, а стрілки – оптимальному маршруту обходу, наведено на рис. 1.1.

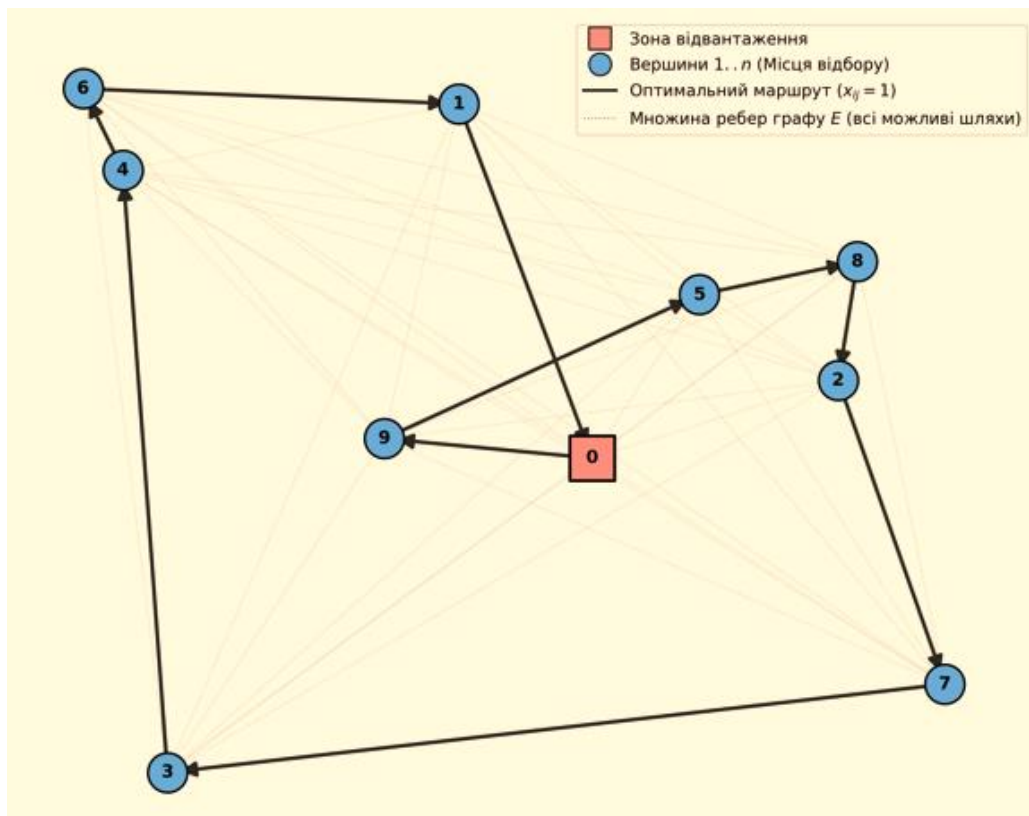


Рисунок 1.1 – Графічна модель задачі маршрутизації з відображенням оптимального шляху

У реальному операційному середовищі завдання набуває додаткової складності та трансформується у **Задачу маршрутизації транспортних засобів** (*Vehicle Routing Problem, VRP*). На відміну від TSP ця модель враховує наявність декількох агентів, таких як комплектувальники чи роботи, та передбачає дотримання низки додаткових умов. До них належать обмеження вантажопідйомності, згідно з якими сумарна вага та об'єм товарів у замовленні не повинні перевищувати місткість візка або навантажувача. Також важливим фактором виступають часові вікна, що визначають необхідність відвантаження замовлення до конкретного терміну [19].

Ці задачі належать до класу **NP-складних** (*NP-hard*), що вказує на неможливість знаходження гарантовано оптимального рішення за поліноміальний час при великій розмірності завдання. Така обчислювальна трудомісткість особливо відчутна при значній кількості **одиниць складського обліку** (*Stock Keeping Unit, SKU*). Саме тому для практичної реалізації у WMS-

системах використовуються не точні методи, такі як повний перебір або метод гілок та меж, а наближені підходи. До них належать евристичні алгоритми, які дозволяють швидко знаходити прийнятне рішення на основі спрощених правил та практичного досвіду. Більш складними є метаврустичні алгоритми, які є стратегіями високого рівня для дослідження простору пошуку та комбiнують різні евристики для уникнення застрягання в локальних оптимумах, тобто попередження передчасної зупинки на рішенні, яке є найкращим лише в обмеженому околі, а не в глобальному масштабі.

**Евристичні та метаврустичні алгоритми.** Необхідність швидкого прийняття рішень у динамічному середовищі складу зумовила популярність біонічних методів оптимізації.

**Генетичні алгоритми** – це клас алгоритмів стохастичного пошуку, що імітують процес природного відбору, де у контексті задачі маршрутизації «хромосома» являє собою закодований варіант маршруту обходу комірок. Цей алгоритм оперує популяцією рішень та ітеративно покращує їх за допомогою трьох ключових операторів. Насамперед застосовується селекція, яка полягає у відборі найкращих маршрутів з найменшою довжиною для подальшого розмноження. Наступним етапом є кросинговер або схрещування, що передбачає обмін частинами маршруту між двома батьківськими рішеннями задля створення нових нащадків. Завершальним оператором виступає мутація, суть якої полягає у випадковій зміні порядку відвідування точок у маршруті, що дозволяє уникнути застрягання у локальних мінімумах. Генетичні алгоритми демонструють особливу ефективність при вирішенні задач VRP, де виникає необхідність одночасної оптимізації розподілу замовлень між комплектувальниками та послідовності їх виконання [20].

**Мурашині алгоритми** базуються на моделюванні поведінки мурах, які знаходять найкоротший шлях від мурашника до їжі, маркуючи його феромонами. У складській логістиці «віртуальні мурахи» прокладають шляхи графом топології складу. Ймовірність вибору наступної вершини (комірки) залежить від евристичної привабливості (близькості) та рівня «феромону»,

який посилюється, якщо цей шлях виявився коротким. Цей метод демонструє високу адаптивність у задачах зі змінною топологією, зокрема за наявності заторів у проходах складу, що дозволяє динамічно перебудовувати маршрути [21].

Підсумовуючи викладене, варто зазначити, що висока адаптивність метаевристичних алгоритмів не здатна повністю компенсувати неефективність складського зонування, спричинену статичністю методів класифікації ABC, XYZ. Оскільки маршрутизація є похідною від фактичного розміщення запасів, пріоритетним завданням стає впровадження інтелектуального групування товарів, що враховує часову динаміку попиту [18].

## **1.2 Адаптація розміщення товарів за допомогою машинного навчання**

Ефективність складської логістики, окрім оптимізації маршрутів переміщення, суттєво залежить від стратегії розміщення запасів. У науковій літературі ця проблема відома як *Storage Location Assignment Problem (SLAP)* або **задача призначення місць зберігання** і полягає у визначенні оптимального місця розташування для кожної товарної одиниці для мінімізації витрат на її подальшу обробку [22]. Ефективне вирішення SLAP дозволяє суттєво скоротити час комплектування замовлень та загальну дистанцію переміщення персоналу і техніки, що є ключовим фактором продуктивності складу. Складність цієї задачі зумовлена необхідністю одночасного врахування множини динамічних факторів, таких як частота обігу товару, його фізичні характеристики та кореляції у попиті між різними позиціями. В рамках концепції Індустрії 4.0 підходи до вирішення SLAP еволюціонують від статичних евристик до динамічних адаптивних моделей [23].

**Від статичного до динамічного розміщення.** Традиційні стратегії розміщення, що використовуються в класичних WMS, зазвичай поділяються на три категорії.

- При **фіксованому закріпленні місць** (*Dedicated Storage*) передбачається, що кожному артикулу присвоюється конкретна комірка. Хоча це спрощує запам'ятовування місць персоналом, такий підхід є вкрай неефективним з точки зору використання простору, оскільки локація залишається порожньою за відсутності товару.
- **Хаотичне зберігання** (*Random Storage*) полягає в тому, що товар розміщується у будь-якій вільній комірці. При такій стратегії максимізується використання об'єму складу, але існує ризик суттєвого збільшення часу комплектування, оскільки популярні артикули можуть опинитися у віддалених зонах.
- **Зональне зберігання** (*Class-based Storage*) є компромісним варіантом, що базується на ABC-аналізі, відповідно до якого товари групи «А» розміщуються у найбільш доступних зонах.

Проте, як було зазначено вище, при статичному зональному зберіганні ігнорується часова динаміка. Застосування методів машинного навчання створює передумови для переходу до динамічного розміщення, де оптимальна локація для товару визначається не жорстким правилом, а ймовірністю його замовлення у найближчий період [24]. Цільову функцію задачі динамічного розміщення можна представити як мінімізацію очікуваних витрат на переміщення:

$$\min \sum_{i=1}^N \sum_{j=1}^M P(d_i|t, X) \cdot C_{ij} \cdot x_{ij}, \quad (1.4)$$

де  $N$  – загальна кількість товарних позицій,  $M$  – кількість доступних місць зберігання (комірок),  $P(d_i|t, X)$  представляє прогнозовану ймовірність попиту на товар  $i$  у момент часу  $t$  за умови вектора ознак  $X$  (сезонність, тренди,



супутні товари), що розраховується моделями ML. Параметр  $C_{ij}$  визначає витрати (час або відстань) на доступ до комірки  $j$  для товару  $i$ , а  $x_{ij}$  є бінарною змінною розміщення, яка дорівнює 1, якщо товар  $i$  розміщено в комірці  $j$ , і 0 в іншому випадку.

Для забезпечення коректності розв'язку та врахування фізичних обмежень складської системи, мінімізація цільової функції (1.4) повинна здійснюватися за умови виконання наступної системи обмежень:

$$\sum_{j=1}^M x_{ij} = 1, \quad \forall i \in \{1, \dots, N\} \quad (1.5)$$

$$\sum_{i=1}^N x_{ij} \leq K_j, \quad \forall j \in \{1, \dots, M\} \quad (1.6)$$

Сформульовані умови мають чітку економічну та фізичну інтерпретацію. Умова (1.5) гарантує, що кожна  $i$ -та товарна позиція буде обов'язково розміщена в одній із доступних комірок складу, унеможливаючи ситуацію, коли товар залишається нерозподіленим у системі обліку. Водночас обмеження (1.6) забезпечує дотримання місткості місць зберігання. Параметр  $K_j$  у цьому виразі визначає максимальну кількість одиниць або об'єм, які може вмістити  $j$ -та комірка. У найпростішому випадку, коли одна локація розрахована на палетомісце або ящик ( $K_j = 1$ ), це обмеження запобігає переповненню місць зберігання та виникненню фізичних колізій при плануванні розміщення.

Отже, сформульована модель (1.4) – (1.6) являє собою задачу лінійного цілочисельного програмування. Враховуючи, що на реальних складах кількість SKU  $N$  та комірок  $M$  може вимірюватися тисячами, простір пошуку рішень зростає експоненціально  $2^{N \times M}$ . Це підтверджує наведену вище тезу про належність подібних задач до класу NP-складних та обґрунтовує необхідність використання евристичних методів або алгоритмів машинного

навчання для знаходження субоптимальних рішень у реальному часі. Включення ймовірнісного компонента  $P(d_i|t, X)$  у цільову функцію дозволяє моделі адаптуватися до змін. Якщо прогнозована ймовірність попиту на товар зростає, алгоритм «переміщує» його у комірки з меншою вартістю доступу  $C_{ij}$  (ближче до зони відвантаження), мінімізуючи загальне математичне сподівання витрат часу на комплектацію.

**Інтеграція штучного інтелекту в системи управління складом.** Сучасний етап розвитку логістики характеризується глибинним впровадженням методів штучного інтелекту у функціонал WMS, адже якщо традиційні системи фокусувалися переважно на обліку та реєстрації транзакцій, то інтелектуальні платформи перебирають на себе функції стратегічного планування та оптимізації [1]. Враховуючи, що програмні комплекси нового покоління генерують значні обсяги даних, а класичні методи часто не адаптовані до динамічних змін профілів замовлень та коливань запасів, виникає об'єктивна необхідність трансформації WMS з інструменту обліку в систему підтримки прийняття рішень. Такий перехід досягається шляхом комплексного застосування трьох основних класів алгоритмів [3]. Методи **навчання з учителем** (*Supervised Learning*), до яких належать штучні нейронні мережі та машини опорних векторів, здатні з високою точністю прогнозувати час виконання складських операцій та навантаження на персонал. Це дозволяє проактивно перерозподіляти ресурси до виникнення проблем з пропускнуою здатністю. Водночас визначальну роль відіграють алгоритми **навчання без учителя** (*Unsupervised Learning*), особливо кластеризація, яка уможливлює виявлення прихованих зв'язків між товарами та оптимізацію їх розміщення на основі подібності попиту. Доповнюють цей інструментарій методи **навчання з підкріпленням** (*Reinforcement Learning*), які дозволяють системі навчатися на власному досвіді, динамічно оптимізуючи маршрути навантажувачів та черги завдань у режимі реального часу для адаптації до непередбачуваних змін у складському середовищі.

Комплексне зіставлення базових параметрів функціонування, яке демонструє еволюційний перехід від статичних процедур до адаптивних алгоритмів, наведено в табл. 1.2.

Таблиця 1.2 – Порівняльна характеристика традиційних та інтелектуальних WMS

Характеристика	Традиційна WMS	Інтелектуальна WMS на основі ІІІ
Основна функція	Реєстрація операцій та контроль залишків	Прогнозування, оптимізація та підтримка прийняття рішень
Принцип розміщення	Статичні правила (ABC, закріплені місця)	Динамічний (спираючись на прогнозні моделі попиту)
Реакція на зміни	Реагування після фактичного настання події	Проактивна (передбачення пікових навантажень)
Обробка даних	Аналіз історичних агрегованих звітів	Обробка Big Data та потокових даних у реальному часі
Методи оптимізації	Статичні алгоритми прийняття рішень	Нейромережі, кластеризація, навчання з підкріпленням

Отже, інтеграція методів машинного навчання, зокрема кластерного аналізу часових рядів попиту, створює необхідні передумови для переходу до гнучких стратегій розміщення, обґрунтовуючи доцільність розробки та впровадження нових моделей кластеризації та оптимізації.

### 1.3 Автоматизація збору товарів, управління запасами та роль ІоТ

Перехід до парадигми «Логістика 4.0» супроводжується трансформацією фізичної інфраструктури складу з пасивного середовища зберігання у складну **кіберфізичну систему** (*Cyber-Physical System, CPS*). Принциповою відмінністю такої екосистеми є безперервна генерація різномірних потоків даних, які описують стан матеріальних активів у режимі реального часу. Технологічним фундаментом цифрової трансформації виступає інтегрований комплекс апаратного забезпечення, який об'єднує

розподілені сенсорні мережі, IoT-пристрої та автономні роботизовані системи. Саме ці засоби формують необхідну інформаційну базу, забезпечуючи накопичення масивів даних високої деталізації, що є важливою передумовою для ефективного застосування методів кластеризації часових рядів.

### 1.3.1 Інтернет речей у складській екосистемі

Технологічним підґрунтям, що забезпечує отримання деталізованих даних про рух товарів, виступає Інтернет речей. Традиційні системи управління складом часто спираються на реєстрацію дискретних подій, наприклад, процес ручного сканування штрих-кодів операторами. Такий підхід призводить до виникнення часових затримок між фактичною зміною фізичного стану системи та оновленням відповідних записів у цифровій базі даних. Подолання цих обмежень стає можливим завдяки впровадженню технологій IoT, які забезпечують перехід до режиму безперервного моніторингу в реальному часі [25].

**Технології автоматичної ідентифікації.** Фундаментальним елементом цієї інфраструктури є технологія **радіочастотної ідентифікації** (*Radio Frequency IDentification, RFID*). На відміну від оптичного розпізнавання, яке базується на послідовному скануванні штрих-кодів, RFID не потребує прямої видимості об'єкта, що забезпечує одночасну ідентифікацію сотень одиниць товарів в зоні дії зчитувача. Конструктивні особливості носія даних, а саме структуру типової пасивної мітки для такого маркування, наведено на рис. 1.2. Завдяки цій особливості стає можливою масова реєстрація вантажів під час проходження через рамки на воротах складу, що скорочує час приймання та відвантаження товарів до кількох секунд. Додатковою перевагою є можливість дистанційної інвентаризації важкодоступних зон автономними дронами, адже здатність RFID-сигналу проникати крізь пакування дозволяє проводити аудит верхніх ярусів без зупинки процесів та залучення підйомної техніки.

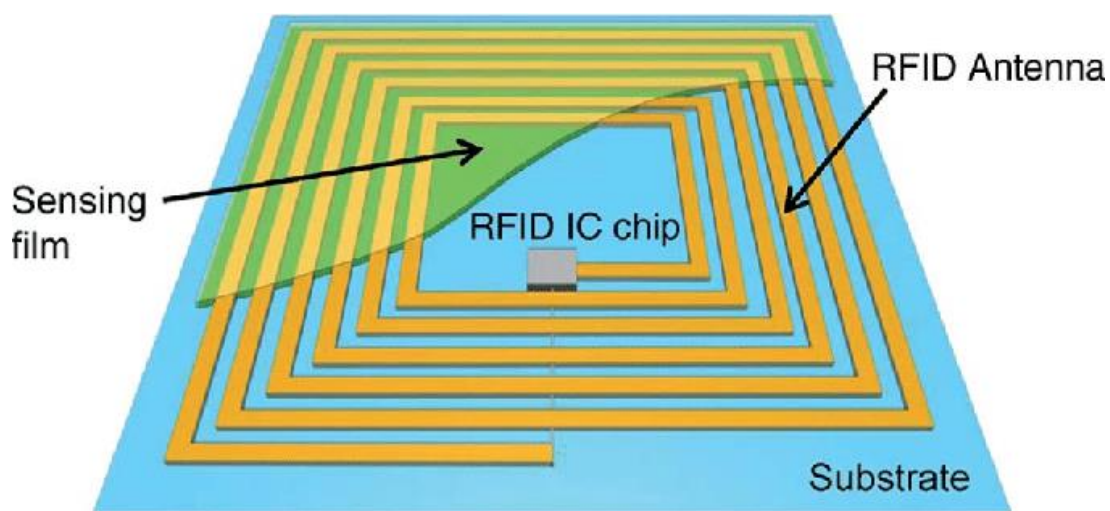


Рисунок 1.2 – Структура пасивної RFID-мітки для маркування товарів  
[26]

Інтеграція RFID-антен безпосередньо в конструкції стелажів формує концепцію **«розумних полиць»** (*Smart Shelves*), візуалізація яких представлена на рис. 1.3. Реалізація зазначеної архітектури фактично перетворює пасивне місце зберігання на активний елемент сенсорної мережі, надаючи WMS можливість не лише відстежувати наявність товару, але й автоматично верифікувати правильність його розміщення в конкретній комірці, миттєво сповіщаючи про помилки комплектування.

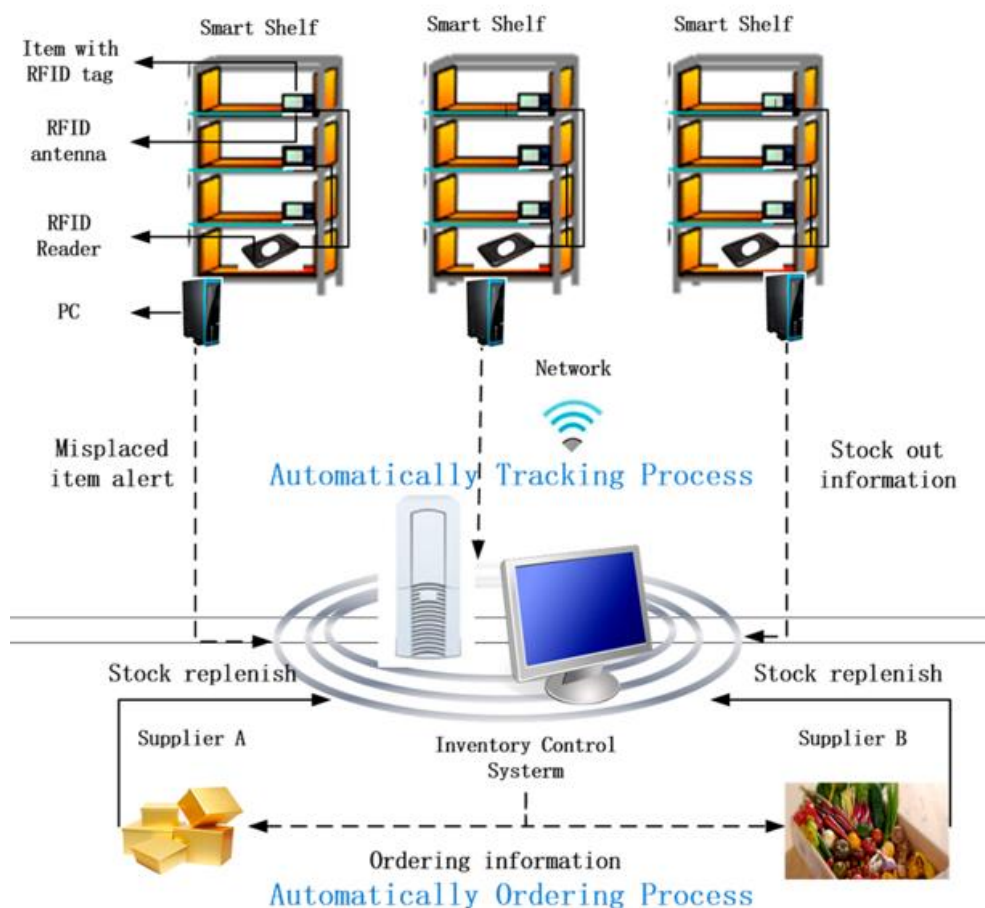


Рисунок 1.3 – Схема функціонування «розумних полиць» [27]

Такі системи, часто доповнені датчиками ваги, здатні автоматично фіксувати момент вилучення або поповнення товару, генеруючи часову мітку події з точністю до мілісекунд. Систематична реєстрація транзакцій формує масиви даних, що описують не лише статичні залишки, а й мікродинаміку попиту, яка є необхідною передумовою для побудови точних прогностичних моделей.

**IoT-контроль параметрів мікроклімату та якості.** Окрім фіксації руху IoT-сенсори забезпечують комплексний та безперервний контроль фізичних параметрів середовища зберігання, що є особливо актуальним для складів зі змішаним асортиментом. Як зазначено в роботі [28], впровадження цих технологій надає можливості для розв'язання проблем, пов'язаних зі зростаючою складністю логістичних операцій. Автори підкреслюють, що завдяки можливостям збору даних у реальному часі та автоматизації багатьох

процесів, IoT стає рушійним фактором підвищення конкурентоспроможності підприємств, дозволяючи не лише відстежувати геолокацію вантажів, але й гарантувати їх якість. Зокрема, використання мультисенсорних датчиків температури, вологості та вібрації дозволяє реалізувати стратегію прогностичного моніторингу для чутливих груп товарів, таких як фармацевтична продукція, харчові продукти та електроніка. Впровадження систем безперервного контролю мінімізує ризики псування продукції внаслідок порушення режимів зберігання ще до моменту відвантаження кінцевому споживачеві. Завдяки цьому відбувається трансформація моделі управління ланцюгом постачання, коли замість ситуативного реагування на інциденти забезпечується випереджальне усунення загроз, що формує надійну технологічну основу для інтеграції високоавтоматизованих роботизованих комплексів.

### 1.3.2 Роботизація процесів та перехід від механізації до автономності

Паралельно з розвитком сенсорних мереж відбувається еволюція виконавчих механізмів складу, що змінює саму філософію комплектування замовлень. Провідним трендом у цій сфері є фундаментальний перехід від традиційної стратегії **«людина до товару»** (*Picker-to-Parts*), яка характеризується значними непродуктивними витратами, до високоефективної моделі **«товар до людини»** (*Parts-to-Picker*). У класичній схемі комплектувальник змушений долати значні відстані між стелажми, витрачаючи, за різними оцінками, до 60 % робочого часу виключно на переміщення, що не додає цінності кінцевому продукту. Натомість впровадження роботизованих систем дозволяє інвертувати цей процес, за якого автоматизовані платформи самостійно доставляють необхідні одиниці зберігання до стаціонарної робочої станції оператора [29]. Така реорганізація технологічного процесу не лише нівелює часові витрати на ходіння, але й

суттєво знижує фізичне навантаження на персонал та мінімізує ймовірність помилок відбору внаслідок втоми.

**Автономні мобільні роботи** (*Autonomous Mobile Robots, AMR*). Сучасні роботизовані рішення базуються на використанні автономних мобільних роботів, які вирізняються здатністю до повністю незалежної навігації у складному середовищі логістичного центру. Функціонування AMR забезпечується **технологією одночасної локалізації і картографування** (*Simultaneous Localization and Mapping, SLAM*). Завдяки інтеграції потоків даних від лідарів, камер глибини та інерціальних датчиків система дозволяє будувати високоточну карту приміщення в реальному часі. Зазначені роботизовані комплекси здійснюють динамічне планування шляху, адаптивно реагуючи на зміни середовища. Вони здатні самостійно оминати перешкоди (навантажувачі, персонал, вантажі) без зупинки виробничого процесу. Така архітектурна гнучкість забезпечує високу масштабованість системи, оскільки додавання нових роботизованих одиниць для покриття пікових навантажень відбувається на програмному рівні і не вимагає капітальних витрат на перепланування фізичного простору логістичного центру. Приклад конструкції мобільного робота-маніпулятора, здатного самостійно вилучати лотки зі стелажів та комплектувати замовлення безпосередньо в проходах складу, наведено на рис. 1.4.





Рисунок 1.4 – Зовнішній вигляд автономного мобільного робота для автоматизованого комплектування замовлень

**Маніпулятори та фізична взаємодія.** Апаратна реалізація автоматизованого відбору часто передбачає використання маніпуляторів з кількома ступенями вільності. В статті [30] розглядається архітектура робота, що поєднує мобільну базу з диференціальним приводом та шестиосьовим маніпулятором. Така система, керована операційною системою Robot Operating System, симуляцію якої реалізовано в тривимірному фізичному середовищі Gazebo, здатна виконувати складні операції захоплення та високоточного переміщення різнотипних вантажів. Використання камер дозволяє роботу розпізнавати об'єкти та коригувати рухи маніпулятора, що є прикладом фізичної реалізації систем, які потребують складного алгоритмічного управління.

Ефективність роботизованої системи можна оцінити через сукупний час виконання замовлення  $T_{order}$ :

$$T_{order} = \sum_{k=1}^K (T_{travel}(d_k) + T_{recognition} + T_{manipulation}), \quad (1.7)$$

де  $K$  – кількість позицій у замовленні,  $T_{travel}$  – час руху, що залежить від дистанції  $d_k$ ,  $T_{recognition}$  – час на розпізнавання об'єкта сенсорами,  $T_{manipulation}$  – час роботи маніпулятора.

### 1.3.3 Роль технічних засобів у формуванні даних

Впровадження розглянутих технологій змінює саму природу даних з якими працює система управління складом, перетворюючи дискретні записи про залишки в безперервний потік подій у реальному часі. Традиційний облік фіксує лише статичний стан запасів на момент інвентаризації. Натомість інтегрована система IoT та роботів генерує багатовимірні часові ряди, що відображають мікродинаміку попиту, коливання робочого навантаження та параметри виконання операцій. Вплив цих засобів на формування інформаційної бази дослідження узагальнено в табл. 1.3. Саме така висока гранулярність та частота оновлення даних створюють необхідну емпіричну основу для застосування методів машинного навчання, дозволяючи виявляти приховані патерни поведінки товарів, які неможливо ідентифікувати класичними методами ABC-аналізу.

Таблиця 1.3 – Вплив технологій IoT та робототехніки на параметри даних для наукового аналізу

Технологічний компонент	Тип генерованих даних	Вплив на операційні процеси	Роль у формуванні бази для дослідження
RFID та Smart Shelves	Часові мітки подій ( $t_{in}$ , $t_{out}$ ), ID товару, локація	Усунення ручного сканування, мінімізація помилок обліку	Забезпечують високу частоту даних, дозволяючи будувати часові ряди попиту високої роздільної здатності
Сенсори середовища	Температура, вологість, освітленість	Забезпечення умов зберігання, запобігання псуванню	Додають параметри обмежень для задач розміщення товарів на основі фізичних властивостей

Автономні роботи	Відстеження маршрутів руху, карта перешкод, час виконання	Реалізація стратегії Parts-to-Picker, зменшення часу $T_{travel}$	Вимагають впровадження алгоритмів оптимізації розміщення товарів для скорочення сумарної довжини маршрутів переміщення робіт
Маніпулятори з шістьма ступенями вільності	Координати захвату, габарити об'єкта, вага	Автоматизація фізичного відбору, безперервність виробничого циклу	Генерують дані про масогабаритні характеристики, що використовуються для кластеризації

Отже, сукупність наведених технологічних факторів та висока варіативність отриманих даних обґрунтовують необхідність переходу до розробки методів кластеризації товарів спираючись на аналіз часових рядів.

#### **1.4 Методологічні основи аналізу даних, часові ряди та формування вибірок**

Впровадження технологій IoT та роботизованих систем призводить до генерації колосальних масивів інформації, проте сама лише їх наявність не гарантує підвищення ефективності складських процесів. Визначальним фактором стає методологія обробки та перетворення дискретних транзакційних записів у впорядковані структури, які придатні для інтелектуального аналізу. Оскільки попит на товари є динамічною величиною, яка змінюється у часі, фундаментальною одиницею аналізу в сучасній логістиці стає не скалярний показник, такий як сумарний обсяг продажів, а векторна структура – часовий ряд.

Враховуючи обмеження статичних методів класифікації запасів, виникає об'єктивна необхідність зміни методологічного підходу до опису попиту, оскільки традиційне оперування усередненими показниками неминуче призводить до втрати інформації про структуру споживання, що

унеможливиює ефективне управління товарами зі складною динамікою. Тому в рамках цього дослідження пропонується перехід до векторного представлення даних, що спрямоване на збереження та аналіз усіх компонентів попиту в їх часовій послідовності. Такий підхід є необхідною передумовою для виявлення прихованих патернів поведінки товарів і створює аналітичне підґрунтя для застосування методів кластеризації та їх адаптації до логістичних задач.

**Критерії якості даних та формування релевантної вибірки.** Ефективність будь-якої моделі машинного навчання прямо корелює з якістю вхідних даних. У реальних умовах функціонування складу інформаційні масиви часто є спотвореними та містять пропуски, дублікати, помилкові записи або аномальні викиди, спричинені технічними збоями. Формування репрезентативної вибірки, що становить основу цільового набору даних (датасету), вимагає чіткої формалізації вимог до інформаційної бази.

Як обґрунтовано у роботах [6, 31] для побудови ефективних моделей оптимізації складських процесів обов'язковим є дотримання низки критеріїв релевантності. Це зумовлено природою цифрової трансформації, що поєднує роботизацію та обробку даних. У такій синергії точність та повнота атрибутів транзакцій є визначальними для успішної оптимізації.

На основі проведеного дослідження [6, 31] систематизовано ключові вимоги до набору даних, необхідних для вирішення задач кластеризації та маршрутизації. По-перше, суттєву роль відіграє часова дискретність, яка забезпечується наявністю точних **часових міток** (*Timestamp Precision*), що вказують на момент оформлення замовлення. Це дозволяє виявляти внутрішньодобову сезонність та пікові навантаження, що є недосяжним при використанні агрегованих звітів. По-друге, транзакційна цілісність даних, представлена через ідентифікатори замовлень (*Order / Invoice ID*), уможливиює проведення аналізу складу споживчого кошика для виявлення товарів, що купуються разом, тобто формують сумісний попит. Це слугує основою для розробки стратегій спільного розміщення. По-третє,

ідентифікація об'єктів через унікальні коди SKU забезпечує відстеження руху конкретних фізичних одиниць, а не товарних груп, що є фундаментом для систем адресного зберігання. Окрім цього, принципове значення мають фізичні атрибути товарів (вага та габарити), які необхідні для перевірки обмежень місткості комірок та розрахунку навантажень на роботизовані системи при плануванні маршрутів. Нарешті, кількісні метрики, які характеризують точний обсяг товарних одиниць, що підлягають обробці в рамках однієї транзакції, визначають частоту поповнення комірок та дозволяють проводити ранжування асортименту за інтенсивністю обігу.

Відповідність визначеним критеріям дає змогу мінімізувати ризики побудови моделей на основі нерелевантних даних та забезпечує можливість переходу до стратегічного прогнозування.

**Джерела даних для експериментальних досліджень.** Через обмеженість доступу до корпоративних даних реальних підприємств, що часто становлять комерційну таємницю та захищені політикою конфіденційності, світова наукова спільнота звертається до альтернативних верифікованих джерел інформації для тестування гіпотез. Такий підхід дозволяє не лише подолати бар'єри доступу до емпіричної бази, але й забезпечити стандартизацію експериментів для об'єктивного порівняння результатів моделювання.

Науково обґрунтованим підходом є використання знеособлених наборів даних з відкритих репозиторіїв таких як Kaggle або UCI Machine Learning Repository. Зазначені масиви даних часто містять реальні історичні логи транзакцій великих ритейлерів і дозволяють відтворювати сценарії поведінки попиту. Зокрема, для верифікації моделей глибокого навчання у сфері електронної комерції застосовується відкритий набір даних «Instacart Market Basket Analysis», який містить деталізовану інформацію про 3 мільйони замовлень від 200 тисяч користувачів та охоплює 134 категорії товарів. У дослідженні [32] цей масив даних був використаний для навчання фрактальної моделі глибокого навчання у поєднанні з алгоритмом оптимізації рою

частинок з метою вдосконалення управління запасами та маршрутизації. Результати моделювання на реальних транзакційних даних продемонстрували зниження витрат на утримання запасів на 20 % шляхом оптимізації видатків на зберігання продукції у складських приміщеннях. Також зафіксовано підвищення точності прогнозування часу доставки замовлень клієнтам на 25 % завдяки врахуванню реальних дорожніх умов та географічних обмежень, що дозволяє нівелювати затримки та оптимізувати використання транспортного парку. Водночас досягнуто покращення ефективності маршрутів на 20 %, що передбачає вибір найбільш оптимальної послідовності відвідування точок доставки. Зазначені показники перевищують результати традиційних методів, до яких автори відносять статичні правила управління та базові статистичні моделі, що характеризуються обмеженою здатністю до адаптації в умовах високої динаміки логістичних процесів.

Іншим показовим прикладом є використання реальних історичних логів онлайн-ритейлера товарів для дому, що охоплюють понад 24 тисячі замовлень за річний період, для вирішення задачі призначення місць зберігання [33]. Автори роботи на основі цього датасету протестували запропонований підхід, що базується на безпосередньому аналізі даних та дозволяє оптимізувати розміщення товарів без попередньої статистичної обробки критеріїв обігу. В основу методології покладено концепцію «хороших пар переміщень», яка передбачає ітеративний пошук та перестановку таких товарних позицій, взаємний обмін місцями зберігання яких гарантує скорочення маршруту відбору товарів. Емпіричні експерименти довели ефективність такого підходу, який дозволив скоротити загальну пройдену відстань комплектувальниками у середньому на 19,54 % та суттєво зменшити час обчислень порівняно з існуючими евристичними алгоритмами кластеризації та асоціацій.

Враховуючи обмеженість наявних експериментальних баз поширеною практикою є застосування синтетичних даних для подолання проблеми недостатності навчальних вибірок або моделювання специфічних операційних умов. У роботі [9] розглядається генерація синтетичних зображень для

навчання систем комп'ютерного зору роботів-комплектувальників, які здійснюють маніпуляції з різнорідними об'єктами, зокрема коробками, пляшками та пакунками довільної форми. Автори застосували варіювання параметрів середовища, а саме: освітлення, кута огляду та фону для створення збалансованих тренувальних наборів, що дозволило ефективно навчити нейронні мережі ідентифікувати об'єкти без необхідності збору та ручної розмітки тисяч реальних фотографій. Така методологія демонструє потенціал використання синтетичної інформації у задачах автоматизації складських процесів, де отримання реальних даних є часозатратним або технічно складним.

Іншим показовим прикладом є дослідження [10] присвячене оптимізації розміщення товарів на основі інтелектуального аналізу даних. Автори використовували синтетично згенеровані транзакції, що імітують асоціативні зв'язки між товарами (споживчі патерни), для верифікації алгоритму із застосуванням правил асоціацій. Результати експериментів на цих даних підтвердили здатність моделі скоротити загальну дистанцію переміщення на 11–37 % порівняно з політикою найближчого вільного місця. Використання синтетичних наборів у цьому випадку дозволило перевірити ефективність алгоритму на різних масштабах даних та структурах замовлень, забезпечуючи надійність висновків перед імплементацією в реальну складську систему.

Отже, системне поєднання глибокого аналізу структури часових рядів із застосуванням розроблених критеріїв якості до формування вибірки формує стійкий методологічний фундамент дослідження. Зазначена методологія не лише гарантує валідність вхідних даних для навчання алгоритмів, але й забезпечує необхідну адаптивність для розробки новітніх моделей кластеризації для оптимізації складських процесів.

## **1.5 Досвід провідних компаній у впровадженні сучасних технологій у складській логістиці**

Інтеграція роботизованих систем у складську логістику змінює підходи не лише до швидкості виконання операцій, а й до стратегій енергоефективності. Традиційно скорочення відстаней переміщення розглядалося крізь призму підвищення продуктивності праці персоналу. Однак для автоматизованого складу цей показник набуває нового змісту. Оптимізація маршрутів є актуальною і для роботизованих агентів, оскільки скорочення дистанцій прямо корелює зі зменшенням енергоспоживання, зниженням зносу механічних вузлів та продовженням терміну служби акумуляторних батарей. Зазначений факт підкреслює універсальність задачі маршрутизації, яка є актуальною як для традиційних технологій з використанням людської праці, так і для повністю автономних систем.

Для розуміння практичної ефективності інтеграції таких рішень доцільно проаналізувати досвід світових лідерів ринку, які вже пройшли шлях від експериментальних впроваджень до масштабування технологій Індустрії 4.0.

### **1.5.1 Кейси глобальних лідерів логістики**

Компанія Amazon після придбання Kiva Systems стала піонером впровадження систем роботизованого відбору, що забезпечують ефективну взаємодію людей та роботів. Зазначена модель «товар до людини» базується на переміщенні цілих стелажів до стаціонарного місця комплектувальника. У фундаментальному дослідженні [11] авторами доведено, що запорукою успіху цієї системи є розробка складних алгоритмів, які забезпечують ефективну синергію між людьми та роботами. Зокрема, автоматизовані агенти перебирають на себе непродуктивні задачі з транспортування вантажів на великі відстані, дозволяючи персоналу сконцентруватися безпосередньо на



процесі відбору, тоді як алгоритмічне керування динамічно оптимізує чергу подачі стелажів, мінімізуючи час простою оператора. Впровадження математичного моделювання та оптимізації процесів дозволило компанії не лише підвищити пропускну здатність складів, а й отримати підтверджений економічний ефект, що оцінюється у мільярди доларів, завдяки точному розрахунку операцій та зниженню витрат на обробку замовлень.

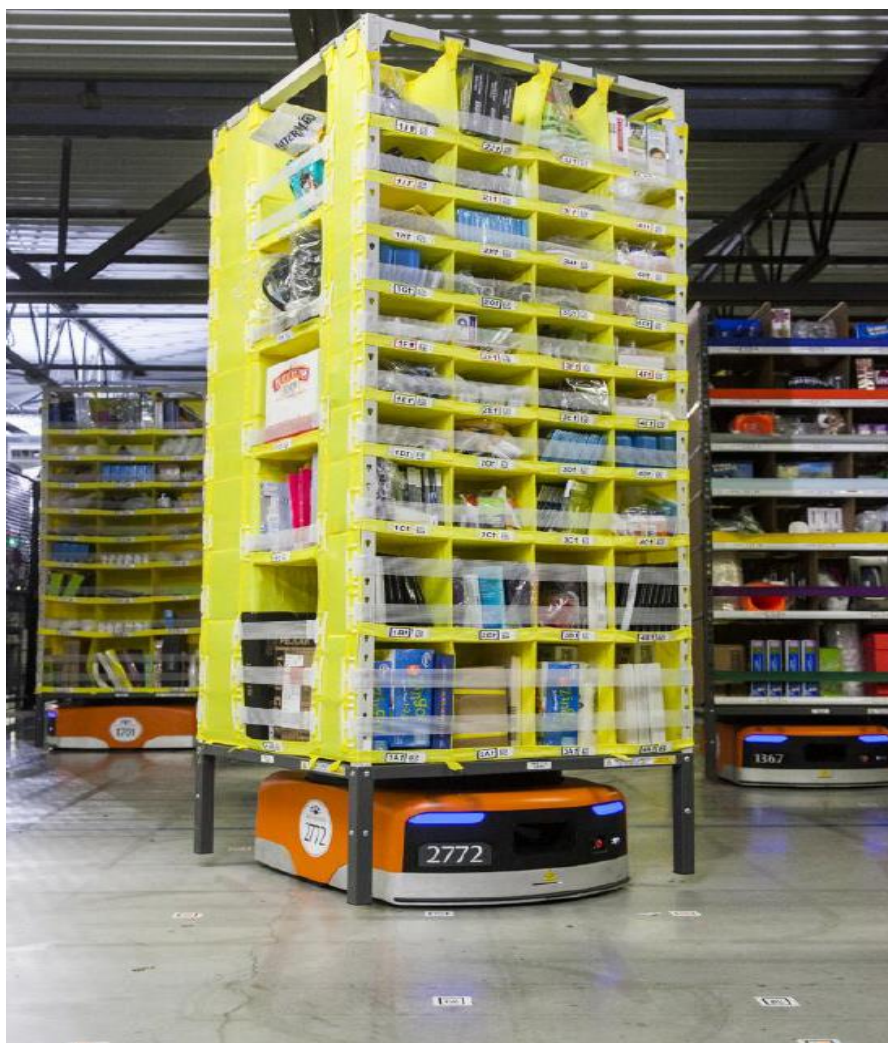


Рисунок 1.5 – Роботизовані платформи Amazon Robotics, що транспортують стелажі до станцій відбору [11]

Китайський гігант Alibaba через свою логістичну платформу Cainiao Network пішов шляхом створення інтелектуальних складських комплексів, де

використовується технологія мультиагентної координації. Як описується в дослідженні [12], зазначені алгоритми керують сотнями мобільних роботів одночасно, дозволяючи їм самоорганізовуватися для оптимального сортування посилок під час пікових навантажень, зокрема 11 листопада в день глобальних щорічних розпродажів. Це демонструє, що фізична наявність роботів є лише інструментом, тоді як реальну ефективність забезпечує програмне ядро управління потоками.

Компанія DHL реалізує стратегію колаборативної робототехніки, інтегруючи автономних мобільних роботів LocusBots, які функціонують у тісній синергії з персоналом. Згідно з даними статті [34] такий підхід дозволив трансформувати роль працівника, суттєво знизивши фізичне навантаження шляхом передачі найбільш виснажливих транспортних операцій роботам. При цьому автоматизовані системи самостійно обчислюють оптимальні маршрути до комірок і очікують там оператора, що не лише нівелює непродуктивні переміщення людей, а й підвищує загальну точність відбору завдяки цифровим підказкам на інтерфейсі робота.



Рисунок 1.6 – Роботи LocusBots, що працюють спільно з персоналом складу

Окремої уваги заслуговує досвід компанії Brightpick, яка впроваджує унікальні рішення для автоматизації фулфілменту. На відміну від класичних автоматично керованих транспортних засобів, що лише возять стелажі, їхній флагманський робот Brightpick Autopicker здатний самостійно відбирати товари безпосередньо в проходах складу, імітуючи дії людини з візком. Це перший у світі комерційний мобільний робот-маніпулятор, який автономно консолідує замовлення (рис. 1.4). Технологія базується на інтеграції вдосконалених систем 3D-зору та генеративного штучного інтелекту, що дозволяє роботу адаптивно розпізнавати та захоплювати об'єкти складної геометричної форми, досягаючи продуктивності, співставної з людською працею [35]. Зазначена інновація свідчить про поступовий перехід від простих транспортних операцій до адаптивної автоматизації, підтверджуючи, що майбутнє складської логістики лежить у площині синергії високої мобільності та інтелектуальних маніпуляцій.

Аналіз наведених кейсів дозволяє стверджувати, що сучасна складська логістика трансформується у кіберфізичну систему, де ефективність залежить від поєднання технологій IoT для збору даних, методів машинного навчання для прогнозування та алгоритмів оптимізації маршрутів для безпосереднього виконання операцій.

Проте навіть найдосконаліші роботи будуть витрачати зайву енергію та час, якщо товари на складі розміщені хаотично або без урахування їхньої сумісності. Тому важливо стає не лише як швидко рухається робот чи людина, а куди саме їм потрібно їхати. Це обґрунтовує необхідність розробки нових методів кластеризації товарів на основі часових рядів та моделей їх розміщення, чому і будуть присвячені наступні розділи дисертації.

## Висновки до розділу 1

У першому розділі дисертаційної роботи проаналізовано сучасний стан алгоритмічного та технологічного забезпечення складської логістики в умовах переходу до концепції Індустрії 4.0. Здійснено детальний аналіз традиційних методів управління запасами, таких як ABC, XYZ та VEN, та виявлено їхню обмеженість, яка полягає у статичності та ігноруванні часової структури попиту, що призводить до неефективного використання ресурсів у динамічному середовищі.

Визначено актуальні наукові виклики, що пов'язані з розв'язанням задач маршрутизації, а саме TSP та VRP, а також призначення місць зберігання SLAP. Встановлено, що ці проблеми належать до класу NP-складних, що зумовлює необхідність застосування евристичних алгоритмів та методів машинного навчання. Проаналізовані технології Інтернету речей та роботизації, зокрема перехід до моделі «Parts-to-Picker», дозволяють отримувати високоточні дані в реальному часі, перетворюючи дискретні транзакції на впорядковані часові ряди.

Спираючись на проведений комплексний аналіз сучасного стану розвитку інтелектуальних систем складської логістики, запропоновано та обґрунтовано доцільність використання часових рядів попиту для кластеризації товарів. Такий підхід створює передумови для виявлення прихованих патернів замовлень продукції, недоступних для класичних методів. Наведене вище підтверджує важливість досліджень у напрямі інтелектуалізації WMS-систем. Основною метою дослідження є розробка нових моделей кластеризації шляхом застосування часових рядів. Це стане підґрунтям для методики Часово-орієнтованого Призначення Місць Зберігання, яка мінімізує сумарні маршрути складських операцій.

## РОЗДІЛ 2. МЕТОДИ КЛАСТЕРИЗАЦІЇ ТОВАРІВ НА ОСНОВІ ЧАСОВИХ РЯДІВ

### 2.1 Побудова часових рядів попиту на товари

Для ефективного аналізу та прогнозування попиту на продукцію необхідно представити історичні дані про попит у вигляді часових рядів. Часовий ряд – це послідовність значень певного показника, упорядкованих за часом зазвичай через рівні проміжки [36]. У нашому випадку таким показником є обсяги продажів або замовлень товарів за певні періоди. Саме спираючись на аналіз минулої динаміки попиту можна виявити закономірності та тренди, що лягають в основу прогнозування майбутніх потреб [37]. У процесі формування бази історичних даних важливими аспектами є вибір частоти даних, визначення джерел для аналізу, врахування сезонності та подій (таких як свята і маркетингові акції), а також вибір відповідних інструментів для обробки [36].

#### 2.1.1 Частота часових рядів попиту

Дані динаміки продажів можуть структуруватися з різною часовою дискретністю, наприклад, добовою, тижневою або місячною. Вибір кроку агрегування даних визначає деталізацію та варіабельність ряду [36]. Розглянемо основні варіанти частоти попиту.

- **Щоденні дані попиту** є найбільш деталізованими, оскільки кожна точка у такому ряді відображає обсяг попиту за окремий день. Це дозволяє виявляти добові коливання та сезонні ефекти в межах тижня, наприклад, зростання попиту в понеділок або зниження у вихідні, а також оперативно реагувати на зміни. Проте щоденні дані зазвичай містять випадкові флуктуації, багато нульових значень і «піки» продажів, особливо для товарів з нерегулярним попитом [36, 37]. Такі ряди мають переривчастий характер, коли більшість

днів попит відсутній, але час від часу з'являються поодинокі великі замовлення, що створює труднощі для прогнозування. Наприклад, якщо товар замовляється раз на два тижні великими партіями, щоденний ряд міститиме переважно нулі та поодинокі стрибки. Для обробки подібних серій можуть використовуватись спеціалізовані методи, такі як метод Кростона, або ж дані агрегуються до ширших часових проміжків на кшталт тижнів чи місяців, що дозволяє згладити випадкові коливання [36, 37].

- **Щотижневі дані попиту.** При агрегуванні замовлень за тижнями кожна точка послідовності відображає сумарний попит за семиденний період. У результаті вплив випадкових щоденних варіацій зменшується, а кількість спостережень у ряді становить близько 52 на рік, що полегшує його візуалізацію та подальший аналіз [36]. Такий рівень підходить для середньострокового планування закупівель або прогнозування загальної динаміки [37]. У щотижневому ряді помітні сталі закономірності на зразок понеділкових замовлень, однак втрачається інформація про розподіл попиту в межах тижня. Якщо попит з'являється лише раз на два тижні, ряд матиме нерівномірність між тижнями, проте з меншою кількістю нульових значень ніж у денному.
- **Щомісячні дані попиту** формуються шляхом агрегування за календарні місяці, у результаті чого отримується 12 точок на рік. Такий рівень деталізації відображає довгострокові тенденції та сезонні коливання, зокрема характерне зростання попиту в грудні [38]. Зменшення частоти істотно знижує шум, адже випадкові флуктуації окремих днів і тижнів нівелюються. Наприклад, за умови двотижневого циклу замовлень у 100 одиниць поперемінно з нульовими тижнями, місячний підсумок на рівні 200 одиниць краще відповідає реальному попиту, ніж щотижнєве середнє значення у 50. Проте короткострокові коливання, як-от знижки або акційні періоди, що тривають лише кілька днів, можуть залишитися непоміченими [36]. Поза тим, для якісного аналізу динаміки на місячному рівні необхідно

мати достатній обсяг історичних даних, бажано щонайменше два-три роки, щоб чітко виявити повторювану річну сезонність [37].

Загалом, вибір частоти залежить від цілей дослідження. Для оперативного управління запасами доцільно застосовувати щоденні ряди, для аналізу середньострокових тенденцій – тижневі або місячні [36, 38].

### 2.1.2 Джерела даних для побудови часових рядів попиту

Відтворення хронології ґрунтується на використанні історичних даних про обсяги продажів або замовлень товарів за певний період. У більшості випадків ці дані зберігаються у внутрішніх інформаційних системах підприємства. Найбільш повним і структурованим джерелом є системи **планування ресурсів підприємства** (*Enterprise Resource Planning, ERP*), в яких фіксуються замовлення, продажі та залишки із точними часовими мітками. Саме ці часові ознаки забезпечують агрегування даних у щоденні, тижневі або місячні інтервали, що робить ERP надзвичайно корисним інструментом для відтворення точної картини змін попиту [39]. Доповненням до ERP виступають WMS-системи, які деталізують етапи виконання замовлень на рівні складу, наприклад, моменти відвантаження. Така інформація може бути критично важливою для аналізу логістичних аспектів попиту [39].

Крім того, важливу роль відіграють системи **управління відносинами з клієнтами** (*Customer Relationship Management, CRM*), платформи інтернет-торгівлі та POS-термінали, які акумулюють замовлення клієнтів із різних каналів. Інформація з таких джерел часто є розподіленою, тому для побудови цілісного часового ряду її необхідно інтегрувати та попередньо очистити. У низці випадків внутрішню інформацію підприємства доцільно доповнювати зовнішніми відкритими даними, такими як статистика споживчого ринку, погодні умови або галузеві тренди. Такі чинники можуть не лише надавати

додаткову інформацію для розуміння попиту, а й виступати незалежними змінними у прогнозних моделях [40].

У ситуаціях, коли історичні дані відсутні або неповні, як-от при запуску нового товару, застосовуються синтетично згенеровані дані. Вони моделюють типові шаблони попиту, такі як сезонність або випадкові флуктуації, і використовуються переважно у дослідницьких цілях. Хоча штучні ряди не є еквівалентом реальних, вони слугують для перевірки працездатності алгоритмів прогнозування в умовах контрольованого середовища [39].

### **2.1.3 Методика побудови часових рядів попиту з історичних даних**

Отримавши вихідні дані про минулі замовлення, необхідно конвертувати їх у формат векторної часової структури. Побудова часового ряду – це процес агрегування та впорядкування даних про попит за вибраними інтервалами часу з отриманням хронологічної послідовності значень. Необхідно забезпечити приведення даних до регулярної часової шкали, зокрема щоденної або щотижневої, без пропусків дат і з правильним їх упорядкуванням [41]. Нижче сформульовано покрокову методику формування часових рядів попиту з використанням історичних даних замовлень.

**1. Збір і підготовка вихідних даних.** Спочатку здійснюється вибірка необхідних даних із джерела – це може бути запит до ERP/WMS, вивантаження CSV-файлу з історією замовлень чи завантаження наявного датасету в аналітичне середовище Python, Excel тощо. На цьому етапі відбирають ключові поля, зокрема дату замовлення, ідентифікатор товару та кількість одиниць, після чого дані готують до агрегування. Очищення даних включає видалення або виправлення дублікатів та помилкових записів, обробку відсутніх значень. Якщо в записах бракує дати, їх слід вилучити або доповнити, а також відфільтрувати дані за відповідним періодом і асортиментом залежно від мети аналізу [41].



2. **Вибір кроку часу та агрегування даних.** Далі визначається крок часу з яким формуватиметься дискретна послідовність залежно від того, чи йдеться про щоденний, щотижневий або щомісячний попит. Наприклад, для щоденного ряду дані групують за датою й найменуванням, підсумовуючи всі замовлення кожної позиції за конкретний день. Якщо протягом дня було кілька замовлень товару, їх підсумовують, а за відсутності замовлень попит дорівнює нулю. Аналогічні операції проводяться для усіх днів в обраному історичному періоді і для кожного товару. Результатом є таблиця з колонками Дата – Товар – Попит в якій кожен запис містить агрегований добовий обсяг попиту на певний товар [42].
3. **Формування хронологічної послідовності.** На цьому етапі агреговані дані впорядковуються за часом для кожного товару, тобто створюється окремий часовий ряд. Впорядкований список дат у межах періоду спостережень, наприклад, за 2022–2023 роки, формують таким чином, щоб до кожної дати прив'язати значення попиту з попереднього кроку. Важливо забезпечити хронологічний порядок спостережень і рівномірну шкалу часу, наприклад, з денним кроком. Саме регулярність і послідовність значень відрізняє часовий ряд від звичайної вибірки, даючи змогу аналізувати залежність даних у часі [41]. На практиці доцільно використовувати інструменти Python, зокрема бібліотеку pandas, яка призначена для ефективної обробки табличних даних. У pandas колонку з датами можна призначити як часовий індекс, що дозволяє здійснювати фільтрацію за датами, ресемплінг (зміну частоти), обчислення ковзних середніх і лагів (зсувів значень ряду назад у часі). Лаги допомагають виявити автокореляцію та формувати вхідні змінні для моделей прогнозування [41, 43]. Тепер часовий ряд для кожного товару можна уявити як таблицю з двома стовпцями в якій перший містить дати, другий – обсяги попиту у відповідні дні.
4. **Обробка пропусків та аномалій.** У реальних даних дні з нульовим попитом після агрегування можуть не фіксуватися через відсутність транзакцій. Пропуски необхідно заповнювати аби зберегти рівномірність

часових інтервалів. Усі відсутні дати додаються до ряду з нульовими значеннями попиту, що забезпечує неперервність і коректну роботу алгоритмів, які потребують однакових інтервалів між спостереженнями. Нулі мають важливе значення, оскільки відображають відсутність попиту й впливають на середні значення та варіативність. Крім пропусків, слід виявити аномальні значення, наприклад, одиничні надзвичайно великі замовлення, які спотворюють тренд. Такі викиди або виключають з аналізу, або коригують, якщо є підстави вважати їх винятковими [41, 42]. У підсумку після виконання кроків 1–4 формуються впорядковані ряди попиту для кожного товару без прогалин і з очищеними аномаліями.

**5. Отримання готового часового ряду.** Після виконання всіх попередніх кроків результатом є повноцінний вектор історії продажів для кожного товару. Якщо аналізується одна конкретна позиція, маємо одномірний ряд, що показує зміну попиту з обраним інтервалом, наприклад, день за днем. У разі залучення багатьох товарів у дослідження формують набір часових рядів по одному для кожного артикулу. Такий формат даних зручний для подальшого аналізу, адже дає змогу будувати графіки попиту в часі, обчислювати показники тренду й варіабельності, а також застосовувати алгоритми прогнозування та кластеризації [43].

Рис. 2.1 ілюструє 60-денний часовий ряд з переривчастим характером попиту, що характеризується переважанням нульових значень і поодинокими сплесками, типовими для товарів з нерегулярним попитом. Це підкреслює важливість заповнення пропущених значень і використання методів згладжування.

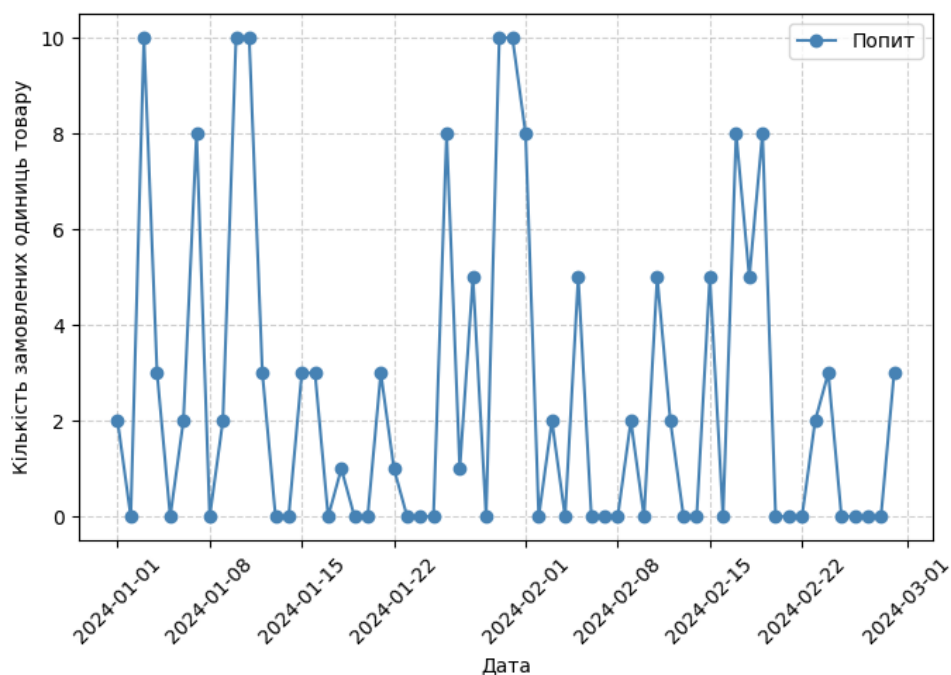


Рисунок 2.1 – Часова послідовність попиту на товар протягом 60 днів (умовні дані)

#### 2.1.4 Облік сезонності, свят і маркетингових кампаній у часових рядах

Попит на товари часто змінюється під впливом сезонності та зовнішніх подій. Сезонність – це регулярні коливання, які повторюються через фіксовані інтервали, наприклад, зростання попиту на подарунки в грудні чи спад продажів зимового одягу влітку. Для коректного прогнозування важливо мати 2–3 повні цикли даних, що дозволяє статистично підтвердити повторюваність патернів. У моделюванні сезонність враховують шляхом введення відповідних індикаторів, як-от змінної місяця або через декомпозицію ряду. Також використовують методи прогнозування з урахуванням періодичних коливань, таких як моделі ARIMA з сезонною компонентою та експоненційне згладжування [44]. У задачах кластеризації спільні сезонні коливання можуть об'єднувати товари з подібним профілем попиту.

Окрім регулярної сезонності попит зазнає впливу нерегулярних, одноразових подій, зокрема святкових періодів та маркетингових кампаній.

Святкові дати часто зумовлюють нетипову поведінку споживачів, спричиняючи як різке зростання попиту, наприклад, напередодні Нового року, так і його спад, зокрема під час тривалих канікул. При цьому характерною є також зміна моменту реалізації попиту, коли покупці здійснюють покупки до або після свята, що призводить до утворення пікових значень або провалів («ям») у часовому ряді. Якщо ці ефекти залишити поза увагою, модель може хибно інтерпретувати їх як частину тренду. Для коректного врахування свят у прогнозних моделях зазвичай застосовують двійкові індикатори, коригувальні коефіцієнти або ж виключають святкові періоди з розрахунків базових значень. У межах цього дослідження вплив свят не було вилучено з вхідних даних, проте за потреби інтеграція календаря свят є нескладним завданням і потенційно може підвищити ефективність класифікації [44].

Що стосується маркетингових кампаній, таких як розпродажі, акції чи рекламні активності, то вони здатні викликати імпульсні сплески продажів на короткий період. Після завершення подібних заходів попит зазвичай повертається до нормального рівня або навіть тимчасово знижується, якщо споживачі здійснили закупівлі наперед. Ігнорування цих ефектів може призвести до переоцінки базового рівня попиту в прогнозних моделях. Відтак у практиці прогнозування рекомендується додавати спеціальні змінні, наприклад, двійкові індикатори періоду акції або включати витрати на маркетинг як незалежні фактори. У задачах кластеризації часового ряду доцільно попередньо нормалізувати дані, зокрема усунути вплив промо-акцій для фокусування на базовому попиті. У цьому дослідженні маркетингові впливи окремо не моделювались, проте усвідомлення їхньої ролі є необхідною умовою для правильної інтерпретації результатів і в практичному застосуванні може сприяти підвищенню точності як прогнозування, так і класифікації [45].

### 2.1.5 Інструменти для побудови та аналізу часових рядів

Процес побудови часових рядів та їх подальшого аналізу неможливий без застосування ефективних програмних і математичних інструментів. Вибір інструментів залежить від обсягу даних, типів необхідних операцій, таких як агрегування, статистичний аналіз, візуалізація чи моделювання, а також від рівня підготовки аналітика. Розглянемо кілька популярних підходів і середовищ, що використовуються для роботи з часовими рядами попиту.

- **Python.** Мова програмування Python є одним із найпотужніших та найпоширеніших засобів аналізу даних, включно з часовими рядами. Python пропонує широку екосистему бібліотек для аналізу часових рядів. Бібліотека pandas забезпечує зручну роботу з таблицями та часовими індексами, а NumPy використовується для високопродуктивних обчислень. Matplotlib і Seaborn дають змогу будувати графіки. Для реалізації класичних статистичних моделей (ARIMA, SARIMA та ETS) застосовується бібліотека statsmodels. Методами машинного навчання, такими як класифікація та кластеризація, оперує scikit-learn та інші спеціалізовані пакети. Фактично Python автоматизує весь цикл аналізу, починаючи із завантаження та агрегування сирих даних і завершуючи побудовою моделей та оцінкою точності результатів. На сьогодні Python вважається одним із найкращих інструментів для аналізу часових рядів завдяки широкому спектру доступних бібліотек та функцій [46].
- **R.** Мова програмування R також є дуже потужним засобом, особливо популярним у наукових і статистичних дослідженнях. Пакети forecast, tseries, zoo, xts, prophet, tsibble та інші забезпечують усі ключові функції серед яких декомпозиція, тестування та побудова моделей. Завдяки великій кількості доступних реалізацій сучасних алгоритмів, R є особливо цінним для академічного прогнозування [47].
- **Microsoft Excel.** Табличний процесор Excel – найбільш доступний і знайомий багатьом інструмент, який, хоч і не спеціалізований під великі

дані, проте широко застосовується на практиці для простих випадків аналізу часових рядів. Він дозволяє агрегувати дані, будувати графіки та застосовувати прості методи прогнозування. Перевагою є доступність і візуалізація, недоліком – обмеженість при обробці великих обсягів та автоматизації [48].

- **Інші інструменти** включають MATLAB, SAS, EViews, SPSS, а також BI-системи, як Power BI та Tableau, які забезпечують візуалізацію та базовий прогноз. У новітніх задачах використовують мову Julia або хмарні сервіси, як-от Google Cloud Forecasting та Azure Time Series. Кожне середовище має свої переваги – від точності алгоритмів до зручності інтеграції у бізнес-процеси [48].

## 2.2. Алгоритми кластеризації часових рядів

У сучасних умовах функціонування логістичних систем, особливо в контексті зростаючої складності процесів управління запасами, дедалі більшого значення набуває аналітика часових рядів попиту. Задля ефективної організації складського зберігання доцільним є групування позицій за подібністю динаміки попиту. Такий підхід оптимізує маршрути переміщення комплектувальників, зменшуючи відстані між точками відбору й тим самим знижуючи операційні витрати та прискорюючи процес збору замовлень [4, 49, 50].

Таке завдання можна реалізувати через кластеризацію часових рядів попиту – методологію, що полягає у виявленні та об'єднанні схожих патернів у тимчасових послідовностях даних. Вона утворює стійкі групи товарів із подібною поведінкою у часі, що є критичним чинником для ухвалення рішень про їх розміщення в межах складського простору [50]. Залежно від структури даних, типу схожості та очікуваної інтерпретованості результатів, можуть використовуватися різноманітні алгоритми кластеризації, які адаптовані для обробки послідовностей.

До основних підходів належать **агломеративна ієрархічна кластеризація** (*Agglomerative Hierarchical clustering, AHC*), метод **k-середніх** (*k-means*), алгоритми кластеризації, що використовують **DTW** (*Dynamic Time Warping*) як міру відстані, а також нейронні моделі, зокрема **самоорганізаційні карти** (*Self-Organizing Map, SOM*) [51, 52]. Кожен із цих підходів має свої переваги й обмеження, які зумовлюють сферу його застосування для управління запасами.

З метою точного опису об'єкта кластеризації розглянемо математичну інтерпретацію часового ряду. Його можна подати як впорядковану послідовність значень у часі:

$$X = \{x_1, x_2, \dots, x_T\}, \quad (2.1)$$

де  $x_t$  – числове значення змінної (наприклад, попиту на товар) у момент часу  $t$ , а  $T$  – загальна кількість спостережень. На відміну від статичних векторів ознак, часові ряди характеризуються внутрішньою структурованістю, а саме: автокореляцією, сезонністю, наявністю аномалій, трендів і можливими зсувами у фазі. Відтак, кластеризація таких даних потребує застосування спеціалізованих методів і метрик подібності, які здатні враховувати особливості часових характеристик.

З огляду на це основними вимогами до алгоритмів кластеризації часових рядів є:

- інваріантність до масштабування та зміщення у часі;
- здатність розпізнавати форму ряду, а не лише абсолютні рівні значень;
- стійкість до шумів і локальних флуктуацій;
- забезпечення високої якості кластерів з точки зору внутрішньої однорідності та зовнішньої відмінності.

З огляду на вищезазначене, кластеризація часових рядів виступає не лише як інструмент аналізу попиту, а й як важливий етап у моделюванні стратегії розміщення товарів на складі, що враховує реальні особливості

поведінки споживачів у часі. Розглянемо ключові алгоритми, що застосовуються для цього завдання, а також наведемо приклади їх використання у практичних сценаріях складської логістики [49, 50].

### 2.2.1 Міри відстані для часових рядів

Обчислення відстані між часовими рядами є фундаментальною операцією в задачах кластеризації та класифікації, оскільки саме поняття «схожості» між рядами визначає формування груп з однаковою динамікою попиту. Від правильного вибору метрики відстані залежить коректність побудови кластерної структури, зокрема при зонуванні складу, де формуються групи товарів з подібним сезонним або циклічним характером попиту [44].

**1. Евклідова відстань (Euclidean Distance)** – найпростіший і найшвидший спосіб порівняння рядів:

$$d_E(X, Y) = \sqrt{\sum_{t=1}^T (x_t - y_t)^2}, \quad (2.2)$$

де  $X = \{x_1, x_2, \dots, x_T\}$ ,  $Y = \{y_1, y_2, \dots, y_T\}$  – два часові ряди однакової довжини.

Цей метод має низьку обчислювальну складність  $O(T)$ , що робить його придатним для великих наборів даних. Проте евклідова метрика є надто чутливою до фазових зсувів та амплітудної різниці між рядами. Тому її застосування до нерегулярних логістичних даних, наприклад, попиту зі сплесками в різні дні тижня часто призводить до помилкових результатів [46].

**2. Алгоритм динамічної трансформації часової шкали (Dynamic Time Warping, DTW)** є більш гнучким підходом, який дає змогу порівнювати ряди з нерівномірною динамікою, фазовими зсувами або відставанням.



$$DTW(X, Y) = \min_W \sqrt{\sum_{(i,j) \in W} (x_i - y_j)^2}, \quad (2.3)$$

де  $W$  – оптимальний шлях вирівнювання, який мінімізує загальну відстань між точками рядів.

DTW є особливо корисним у складських системах у випадках, коли товари мають подібну структуру попиту, але відрізняються циклічним зміщенням. Наприклад, попит на шоколад зростає наприкінці кожного місяця, тоді як на каву – на його початку. Основним недоліком методу є висока обчислювальна складність  $O(T^2)$ , яка суттєво ускладнює його застосування для аналізу великих обсягів даних, зокрема десятків тисяч товарних позицій. Це обмеження можна подолати за допомогою спеціальних оптимізацій, таких як FastDTW – алгоритму наближеної оцінки DTW з лінійною складністю, або шляхом застосування обмеженого вікна пошуку, зокрема методу Sakoe-Chiba Band [44].

**3. Shape-Based Distance (SBD)** – орієнтована на форму метрика, яка ґрунтується на **нормалізованій крос-кореляції** (*normalized cross-correlation*) між рядами:

$$d_{SBD}(X, Y) = 1 - \max_{\tau} NCC(X, Y^{\tau}), \quad (2.4)$$

де  $Y^{\tau}$  – циклічно зсунута версія ряду  $Y$  на  $\tau$  позицій, а  $NCC$  – нормалізована крос-кореляція.

Метрика використовується в алгоритмі k-середніх і дозволяє виявляти ряди з однаковою «формою» (патерном), навіть якщо абсолютні значення або часові масштаби відрізняються. Це робить її вельми корисною в логістичних задачах, пов'язаних із попитом на групи товарів, чий тренд має схожу траєкторію в часі незалежно від конкретних дат [46].

### 2.2.2 Агломеративна ієрархічна кластеризація

Агломеративна ієрархічна кластеризація належить до класичних і широко застосовуваних методів побудови кластерів, заснованих на попарних відстанях між об'єктами [53]. У її основі лежить ідея послідовного об'єднання об'єктів або кластерів на підставі мінімальної відстані між ними, що ілюструється у вигляді дендрограми, яка відображає ієрархічну структуру агломерації, де кожен рівень відповідає певному етапу злиття. Такий підхід дозволяє не лише гнучко моделювати ієрархічну структуру даних, але й суттєво підвищує інтерпретованість результатів, що особливо важливо у контексті логістики, де структура попиту часто є варіативною та сезонною [43].

На відміну від методів, які вимагають попереднього задання кількості кластерів, АНС генерує кластерну структуру без заздалегідь відомої кількості груп, а рішення про їх число приймається на основі аналізу дендрограми. Це забезпечує особливо високу гнучкість при роботі з неоднорідними або складними даними, зокрема у випадку аналізу історичних трендів [43, 53].

Для візуалізації результатів АНС зазвичай використовують дендрограми, які графічно відображають послідовність об'єднань об'єктів у кластери відповідно до обчислених відстаней. На рис. 2.2 наведено приклад такої дендрограми, побудованої на основі 100 синтетичних об'єктів, згенерованих як точки в багатовимірному просторі з наперед заданою кластерною структурою. Зображена горизонтальна червона штрихова лінія позначає відсікання на рівні 50 умовних одиниць, що використовується як порогове значення. Саме на цьому рівні було здійснено поділ дендрограми, що дозволило виділити три окремі кластери. З аналізу дендрограми можна зробити висновок, що верхні гілки дерева представляють віддаленіші об'єднання, тоді як нижчі – більш подібні об'єкти або малі групи з низькою внутрішньою варіацією. Візуально помітно, що кластери добре розділені вертикальними стовпцями, які відходять від порогового рівня, і кожна група

утворює окрему гілку, що свідчить про їхню внутрішню узгодженість і чітку міжкластерну відмінність. Крім того, дендрограма дає змогу оцінити стабільність кластерної структури, оскільки великі вертикальні відстані між рівнями злиття свідчать про суттєву відмінність між об'єднуваними групами [53]. Це є цінною властивістю при формуванні зон зберігання на складі, де групи товарів мають подібні часові патерни попиту.

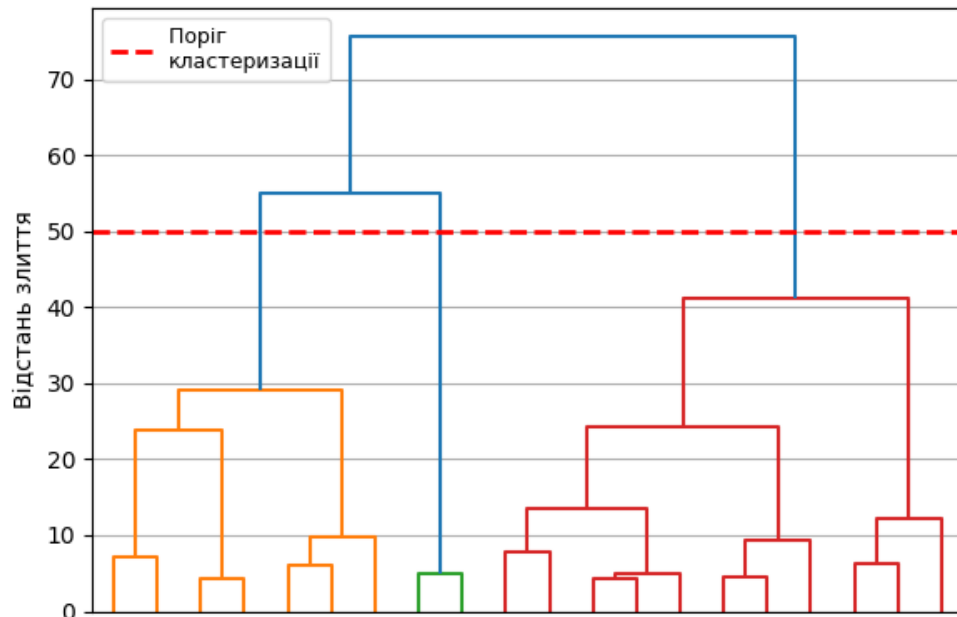


Рисунок 2.2 – Дендрограма агломеративної кластеризації

Загальна процедура агломеративної кластеризації включає такі етапи:

1. На початковому етапі кожен часовий ряд розглядається як окремий кластер.
2. Обчислення міжкластерних відстаней на основі заздалегідь обраної метрики (наприклад, DTW або SBD).
3. Злиття двох найближчих кластерів відповідно до обраного **критерію зв'язності** (*linkage*).
4. Повторення кроків 2–3 до тих пір, поки не буде сформовано один кластер або не досягнуто порогової відстані.

На третьому етапі ключову роль відіграє схема зв'язності, яка визначає, як саме розраховується відстань між кластерами. Залежно від обраного

критерію, міжкластерна відстань визначається за однією з наведених нижче формул:

– **мінімальна відстань між елементами кластерів** (*single linkage*):

$$d(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y) \quad (2.5)$$

– **максимальна відстань** (*complete linkage*):

$$d(C_i, C_j) = \max_{x \in C_i, y \in C_j} d(x, y) \quad (2.6)$$

– **середня попарна відстань** (*average linkage*):

$$d(C_i, C_j) = \frac{1}{|C_i||C_j|} \sum_{x \in C_i} \sum_{y \in C_j} d(x, y) \quad (2.7)$$

Злиття кластерів триває доти, доки міжкластерна відстань не перевищить заданий поріг  $\Theta$ , тобто виконується умова:

$$d(C_i, C_j) < \Theta \quad (2.8)$$

АНС разом із метриками DTW або SBD демонструє високу ефективність при кластеризації часових рядів, що мають складну або нерегулярну структуру попиту [43, 53]. Такий підхід є доцільним у ситуаціях, коли товари характеризуються сезонністю, циклічністю або значними локальними флуктуаціями. Наприклад, це можуть бути напої, побутова хімія або продукти харчування попит на які змінюється залежно від місяця, погоди або святкових періодів. У цьому контексті АНС виокремлює групи товарів із подібною динамікою, що є надзвичайно корисним у процесах прогнозування попиту, формування асортименту або оптимізації логістичних рішень [43].

Основною перевагою агломеративної ієрархічної кластеризації є її незалежність від попереднього задання кількості кластерів, що робить метод

зручним у задачах де кількість груп невідома наперед. До того ж, АНС дозволяє побудувати ієрархічну структуру подібності, яка гнучко поєднується з метриками DTW, SBD або Euclidean [53]. Завдяки цьому аналітик отримує ширші можливості адаптації методу до особливостей даних.

Ще однією важливою перевагою методу є його висока інтерпретованість, яка досягається шляхом візуалізації результатів кластеризації у вигляді дендрограми. Така дендрограма графічно демонструє послідовність злиття кластерів і дозволяє не лише оцінити структуру подібності, а й обґрунтовано вибрати кількість кластерів на основі порогової відстані, визначеної як висота об'єднання гілок дерева [53].

Водночас, попри наведені переваги, метод має суттєві обмеження. Насамперед, ієрархічна кластеризація характеризується високою обчислювальною складністю, яка при використанні повної матриці відстаней зростає до  $O(n^2)$ , де  $n$  позначає кількість об'єктів, таких як товари або їхні історії продажів [53]. Такий рівень трудомісткості суттєво перешкоджає масштабуванню методу на великі обсяги даних, наприклад, при аналізі десятків тисяч позицій на складі. Поза тим, щоразу при новому злитті необхідно оновлювати матрицю попарних відстаней, що потребує значних обсягів оперативної пам'яті.

У зв'язку з цим на практиці АНС рекомендується застосовувати на обмежених підмножинах даних, зокрема для попереднього аналізу структури попиту в середніх за розміром групах товарів (наприклад, 500–1000 SKU). На основі отриманих висновків аналітик може прийняти рішення щодо застосування спрощених або швидших методів кластеризації до повного масиву.

### **2.2.3 Кластеризація методом k–середніх**

Алгоритм k-середніх у своїй класичній формі не враховує особливостей часової залежності, проте його адаптація для кластеризації таких даних

вимагає відповідних модифікацій, зокрема з урахуванням форми тимчасових патернів, а не лише абсолютних значень. Одним із поширених підходів є використання метрики SBD, що ґрунтується на нормалізованій крос-кореляції та забезпечує коректне зіставлення рядів незалежно від їх амплітуди, масштабування чи часових зсувів [55].

Формула обчислення центроїда (представника кластера) в алгоритмі  $k$ -середніх має наступний вигляд:

$$c_k = \arg \max_c \sum_{x \in C_k} NCC(x, c), \quad (2.9)$$

де  $C_k$  – множина часових рядів, що належать до кластера  $k$ ,  $NCC(x, c)$  – нормалізована крос-кореляція між часовим рядом  $x$  і центром  $c$  [54].

Під час ітераційного оновлення центроїди алгоритму  $k$ -середніх формуються не як середні арифметичні, а як оптимальні структури, що максимально геометрично подібні до всіх часових рядів у кластері.

Завдяки використанню нормалізації, алгоритм демонструє інваріантність до масштабу, що критично важливо у логістичних задачах. Наприклад, товари з однаковим циклічним шаблоном, але з різними обсягами продажів (попит у літрах або штуках), можуть бути правильно згруповані лише на основі форми їх динаміки [55].

Для сегментації великого асортименту метод  $k$ -середніх є надзвичайно корисним для виявлення груп товарів із синхронізованими коливаннями попиту. Наприклад, аналітика може виявити, що дитячі іграшки, теплі напої та декорації мають схожий сезонний попит перед святами. Це дозволяє об'єднати їх у загальний кластер для стратегічного розміщення у зоні швидкого доступу в межах складу.

Додатковою перевагою методу  $k$ -середніх є висока стабільність результатів, що забезпечується збереженням форми часових рядів завдяки використанню нормалізованих метрик. Крім того, однією з істотних переваг є

порівняно висока швидкодія алгоритму, складність якого оцінюється як приблизно  $O(nkT \log T)$ , де  $T$  позначає довжину ряду, а  $k$  – кількість кластерів [54]. Також важливим є те, що метод не потребує попереднього масштабування вхідних даних, оскільки нормалізація вбудована безпосередньо у метрику порівняння [55].

Разом із тим, як і більшість варіантів алгоритму  $k$ -середніх, метод вимагає заздалегідь фіксованої кількості кластерів  $k$ , що може бути проблематичним у випадках, коли структура даних наперед невідома. Додатковим ускладненням є потенційна чутливість результатів до початкової ініціалізації центрів кластерів. Тому на практиці часто застосовується багаторазовий запуск алгоритму з подальшим вибором найкращого кластерного розбиття за внутрішнім критерієм якості, наприклад, за допомогою коефіцієнта силуету або показника суми внутрішньокластерних відстаней [55].

Ключовими перевагами методу є висока чутливість до геометричної форми ряду, інваріантність до зміщення та масштабу, а також хороша придатність до роботи з великими обсягами даних. Попри це, обмеженням залишається потреба в попередньому заданні кількості кластерів, а також ймовірна залежність результатів кластеризації від початкових умов, що впливає на їхню стабільність.

Узагальнюючи викладене, можна зробити висновок, що метод  $k$ -середніх є особливо ефективним інструментом для кластеризації часових рядів із регулярною поведінкою. У логістичних системах його застосування забезпечує точніше визначення подібних за динамікою товарів, орієнтуючись не лише на абсолютні значення, а й на схожість сезонного профілю, що підвищує адаптивність до змін попиту та сприяє прийняттю більш обґрунтованих управлінських рішень [55].

### 2.2.4 Кластеризація на основі DTW + k-medoids

Стандартний алгоритм k-means не придатний до використання з метрикою DTW, оскільки він ґрунтується на обчисленні середнього значення, що не визначене у просторі часових рядів з довільними вирівнюваннями. Проте існує адаптація цього підходу – алгоритм кластеризації **к-медоїдів** (*k-medoids*), де замість обчислення середнього центроїда використовується реальний представник – медоїд, тобто такий часовий ряд із кластера, що мінімізує суму відстаней DTW до решти об'єктів [56, 57].

Формально, центр  $c_k$  для кластера  $C_k$  визначається як

$$c_k = \arg \min_{x \in C_k} \sum_{y \in C_k} DTW(x, y) \quad (2.10)$$

Така модифікація дозволяє використовувати DTW як метрику подібності, при цьому зберігаючи ідею прототипного представника кластера. У деяких реалізаціях використовується також Soft-DTW – диференційована версія DTW, що уможливлює застосування градієнтних методів оптимізації та робить можливим використання навіть модифікованих k-means [58].

В умовах логістичної практики метод DTW + k-medoids виявляється корисним для кластеризації товарів з високою нерегулярністю попиту, зокрема у разі, коли форма шаблону зберігається, але спостерігаються зміщення у фазі, наприклад, акційні товари попит на які зростає в окремі проміжки часу залежно від регіону. Завдяки такій гнучкості алгоритм ефективно групує разом ті позиції, які не мають жорсткої сезонності, але загальна логіка попиту є подібною. Ключова перевага – здатність обробляти часові зсуви, стискання та розтягування шаблонів, що є типовими для реальної статистики продажів. Наприклад, усілякі варіанти пакування того самого товару можуть демонструвати схожий попит, але зі зміщенням у часі через асортиментну політику постачальника [57].



Проте основним обмеженням цього підходу є висока обчислювальна складність, особливо при побудові повної DTW-матриці попарних відстаней. При обробці великої кількості товарних позицій (тисячі або десятки тисяч SKU), це може створювати суттєве навантаження на апаратні ресурси. При використанні класичної метрики DTW обчислювальна складність зіставлення лише однієї пари об'єктів становить  $O(T^2)$ , що зумовлює квадратичне масштабування повної матриці відстаней  $O(n^2 T^2)$ , де  $n$  – кількість рядів, а  $T$  – довжина кожного ряду. Така складність робить застосування методу в умовах великої кількості об'єктів обчислювально витратним, тому на практиці доцільним є впровадження прискорювальних стратегій. До таких стратегій зазвичай належать обмеження простору пошуку за допомогою вирівнювального вікна, зокрема методу Sakoe–Chiba Band, а також використання наближених варіантів алгоритму, таких як FastDTW. Крім того, застосовують попереднє фільтрування часових рядів за простішими статистичними характеристиками, наприклад середнім значенням чи стандартним відхиленням [59].

Ключовою перевагою підходу DTW + k-medoids є здатність природно враховувати фазові зсуви, масштабні відмінності та варіації в часовій структурі даних, що робить його особливо гнучким у роботі з нерегулярними шаблонами попиту. Поза тим, він демонструє високу точність кластеризації навіть у складних реальних сценаріях із сезонними коливаннями або нетиповою динамікою [57].

Водночас метод DTW + k-medoids має і низку недоліків. По-перше, він передбачає високі обчислювальні витрати при зростанні кількості об'єктів. По-друге, вимагається попереднє визначення кількості кластерів  $k$ , що не завжди є очевидним при роботі з реальними даними. По-третє, масштабування алгоритму без застосування оптимізаційних технік є доволі складним [59].

Підсумовуючи, можна стверджувати, що підхід DTW + k-medoids є потужним аналітичним інструментом для логістичних та складських задач, зокрема в умовах непередбачуваного або хаотичного попиту. Його орієнтація

на детальну сегментацію SKU дозволяє підвищити точність планування, адаптувати стратегії поповнення запасів та вдосконалити управління складськими ресурсами.

### 2.2.5 Самоорганізаційна карта (SOM)

Алгоритм самоорганізаційних карт, запропонований Теуво Кохоненом, є ефективним нейромережевим підходом для кластеризації та візуалізації високовимірних даних, зокрема часових рядів. SOM реалізує механізм проєкції багатовимірнього простору у двовимірну сітку із збереженням топології, за якої подібні об'єкти автоматично розміщуються поруч один з одним [60]. На відміну від класичних методів кластеризації, SOM не базується на жорстко визначених центроїдах або глобальних критеріях подібності між усіма об'єктами. Натомість він використовує конкурентний механізм навчання нейронів, що адаптує ваги вузлів карти на основі схожості з вхідними векторами [61].

Математична суть алгоритму полягає в тому, що для кожного вхідного вектора-зразка  $x$  обирається нейрон  $\omega_i$  вага якого є найближчою до  $x$  за евклідовою метрикою.

$$c = \arg \min_i \|x - \omega_i\| \quad (2.11)$$

Після цього виконується оновлення ваг не лише для «переможця»  $\omega_i$ , а й для його сусідів, визначених за допомогою функції околу. Такий підхід поступово формує карту, де кожен вузол (нейрон) відповідає певному типу часової динаміки, забезпечуючи локальне узгодження вхідного простору [61].

У контексті логістичних задач SOM розпізнає характерні патерни споживання різної складності (стабільні, сезонні, хаотичні), не потребуючи попереднього задання кількості кластерів. Це робить метод особливо придатним для обробки великих масивів інформації, таких як дані щодо попиту на тисячі SKU. SOM також відіграє важливу роль у стратегічному

прийнятті рішень щодо зонування складу, прогнозування або сегментації поведінки споживачів [60].

Ключовою перевагою алгоритму самоорганізаційної карти є його здатність до топологічного збереження. Ця властивість гарантує, що часові ряди з подібною динамікою проєктуються у сусідні зони двовимірної сітки [61]. В результаті формуються візуально компактні області, які характеризуються внутрішньою однорідністю поведінки, що суттєво полегшує візуалізацію та подальший аналіз результатів. Зокрема, це створює умови для інтерактивної інтерпретації кластерної структури, наприклад, шляхом побудови теплових карт активності або виділення аномальних вузлів чия поведінка значно відрізняється від загальної [60].

Основними перевагами методу SOM є висока наочність і пояснюваність результатів. Завдяки збереженню топології вхідного простору, алгоритм ефективно відтворює локальні залежності між об'єктами. До того ж, SOM не вимагає попереднього визначення кількості кластерів, що є важливою перевагою у випадках, коли структура даних заздалегідь невідома [61].

Разом із тим метод має і низку обмежень. По-перше, його результативність істотно залежить від правильного налаштування гіперпараметрів, зокрема розміру сітки, функції сусідства та кроку навчання. По-друге, для досягнення коректних результатів необхідно виконувати попередню нормалізацію та масштабування вхідних даних. По-третє, кластерна структура, сформована SOM, зазвичай має нечітко окреслені межі, що може потребувати додаткової обробки, наприклад, постпроцесингової сегментації мапи з використанням інших методів [61].

Окрему увагу слід приділити вибору розміру карти, оскільки цей параметр значною мірою впливає на якість кластеризації. Недостатньо велика кількість вузлів призводить до надмірної агрегації об'єктів, внаслідок чого втрачається деталізація та глибина аналізу. Натомість надмірно велика сітка може ускладнити узагальнення та призвести до перенавчання [61]. У практичних задачах, пов'язаних з аналізом товарних потоків або управлінні

запасами, SOM зазвичай застосовується на завершальних етапах аналітичного процесу, тобто після виконання попередньої кластеризації, фільтрації або зменшення розмірності. У такому контексті цей алгоритм виступає як інструмент виявлення прихованих патернів, забезпечуючи наочну підтримку процесу прийняття рішень [60].

### 2.2.6 Порівняльна характеристика алгоритмів

У табл. 2.1 подано узагальнену характеристику розглянутих алгоритмів кластеризації часових рядів відповідно до таких критеріїв:

- стійкість до зсуву у часі – здатність алгоритму виявляти схожість між рядами зі зміщенням фаз;
- інтерпретованість – легкість тлумачення результатів кластеризації з точки зору прийняття управлінських рішень;
- масштабованість – здатність алгоритму ефективно працювати з великими наборами даних;
- критерій «необхідність у параметрі  $k$ » визначає чи слід попередньо встановлювати кількість кластерів.

Таблиця 2.1 – Порівняльна характеристика алгоритмів кластеризації часових рядів

Алгоритм	Стійкість до зсуву у часі	Інтерпретованість	Масштабованість	Необхідність у параметрі $k$
ANC + DTW	Високий	Високий	Низький	Ні
$k$ -середніх	Середній	Середній	Середній	Так
DTW + $k$ -medoids	Високий	Високий	Низький	Так
SOM	Низький	Високий	Середній	Ні

Вербальні рівні «Високий», «Середній», «Низький» використовуються для відносного порівняння алгоритмів кластеризації за відповідністю кожному

з критеріїв. Такий підхід полегшує інтерпретацію та пришвидшує ідентифікацію сильних та слабких сторін кожного з методів.

На підставі проведеного порівняльного аналізу алгоритмів кластеризації часових рядів можна сформулювати низку узагальнених рекомендацій, що допомагають обґрунтовано обирати відповідний метод залежно від характеру задачі. Зокрема, для точного моделювання сезонних коливань або зсувів у часі доцільно обирати методи на основі метрики DTW. Вона забезпечує природну інваріантність до фазових зрушень. Якщо ж ключовим критерієм виступає швидкість обробки великих обсягів даних, найбільш придатними є алгоритми  $k$ -середніх або самоорганізаційні карти, які характеризуються високою масштабованістю та спрощеною інтерпретацією результатів. Водночас для задач глибокого ієрархічного аналізу структурної подібності між часовими рядами доцільно застосовувати агломеративну ієрархічну кластеризацію в поєднанні з метриками DTW або SBD. Такий підхід забезпечує високу точність групування навіть у складних сценаріях.

### **2.3. Вибір оптимального кластерного розбиття для цілей зберігання**

Одним із ключових етапів кластерного аналізу є визначення оптимальної кількості кластерів  $k$ . Вибір  $k$  безпосередньо впливає на практичну придатність результатів, оскільки у логістиці кількість кластерів може відповідати кількості зон зберігання чи категорій товарів на які поділяється склад. Наприклад, при зонуванні складу часто припускають, що число виділених зон дорівнює числу кластерів, отриманих при групуванні SKU за схожістю попиту. Таким чином, неправильний вибір  $k$  може призвести до неефективного розміщення, адже занадто мала кількість кластерів об'єднає надто різноманітні товари і втраяться деталі попиту, тоді як занадто велика кількість спричинить дроблення товарних груп і ускладнить управління. Кластеризаційні алгоритми на кшталт  $k$ -середніх потребують заздалегідь заданого  $k$ , що є проблемою, коли структура даних невідома. У науковій

спільноті запропоновано низку методів для обґрунтування вибору оптимального числа кластерів, зокрема **метод ліктя** (*elbow method*), що базується на **внутрішньокластерній сумі квадратів** (*Within-Cluster Sum of Squares, WCSS*), **коефіцієнт силуету** (*silhouette score*), **gar-статистика**, **індекс Девіса–Болдуїна** (*Davies-Bouldin Index*) тощо [62]. Нижче розглянуто ці підходи, їх придатність у логістичних завданнях та комбінування з бізнес-метриками.

**Метод ліктя.** Цей підхід полягає в аналізі залежності внутрішньокластерної дисперсії (суми квадратів відхилень точок від центрів кластерів) від кількості кластерів  $k$ . Ідея полягає в тому, що зі збільшенням  $k$  внутрішньогрупова варіація зменшується, але після певного моменту виграш від додавання кластерів різко сповільнюється. На графіку «WCSS vs.  $k$ » шукають характерний злам (подібний до «ліктя»), після якого крива вирівнюється. Значення  $k$  в точці цього згину і вважається обґрунтованим вибором оптимального числа кластерів. Цей метод є інтуїтивно простим адже аналітик обчислює кластеризацію для діапазону значень  $k$ , наприклад, від 1 до 60 і будує криву  $WCSS(k)$ , після чого візуально визначає «лікоть». Для оптимізації топології складу це дозволяє визначити скільки кластерів товарів доцільно виділити аби значно знизити внутрішньогрупову варіативність попиту [63-65].

На рис. 2.3 зображено умовний приклад elbow-діаграми для штучно згенерованого набору SKU. Видно, що значення WCSS поступово зменшується зі збільшенням кількості кластерів  $k$  від 1 до 60. Найбільш виражене зниження спостерігається на початку, після чого темп зменшення дисперсії сповільнюється. В області  $k \approx 8$  спостерігається «лікоть» – подальше збільшення числа кластерів призводить до незначного виграшу у внутрішньогруповій однорідності. Отже, значення  $k = 8$  можна вважати обґрунтованим вибором.

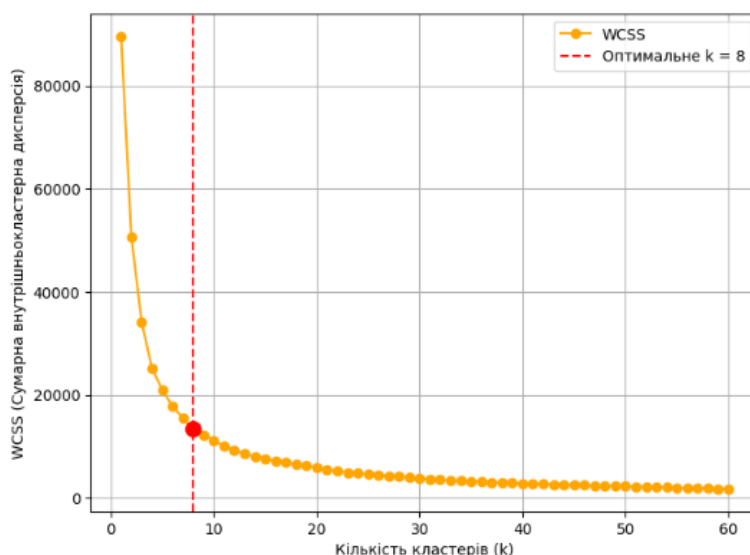


Рисунок 2.3 – Elbow-діаграма для визначення оптимального числа кластерів

Метод ліктя має перевагу в тому, що спирається на просту метрику компактності кластерів і не потребує складних обчислень, оскільки для його застосування достатньо результатів самого алгоритму кластеризації. Водночас його недоліком є суб'єктивність, адже крива WCSS не завжди має чітко виражений злам, тому окремі фахівці можуть по-різному інтерпретувати оптимум. Існують і формальні підходи для автоматизації пошуку «ліктя», наприклад, коли проводять пряму через першу і останню точки графіка та знаходять кластер, для якого відстань від кривої WCSS до цієї прямої є максимальною [66]. Такий геометричний критерій, запропонований в [67] дозволяє алгоритмічно визначити злам і уникнути суб'єктивної оцінки. У контексті складу метод elbow є корисним стартовим інструментом, оскільки дає перше наближення до обґрунтованої кількості кластерів на які варто поділити товари, щоб зменшити внутрішньогрупові варіації попиту. Надалі цей вибір може бути уточнений іншими метриками.

**Коефіцієнт силуету.** Альтернативою візуальному методу ліктя є кількісна оцінка якості кластеризації за допомогою коефіцієнта силуету [63, 64]. Силует для окремого об'єкта  $i$  визначається як різниця між середньою відстанню від  $i$  до точок власного кластера ( $a_i$ ) та мінімальною середньою

відстанню від  $i$  до точок сусіднього кластера ( $b_i$ ), нормована максимальним із цих двох значень:  $s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$ . Значення  $s_i$  варіюється від -1 до +1. Показник, близький до +1, свідчить про якісну кластеризацію. Натомість значення біля нуля вказує на граничний стан об'єкта, а від'ємне – на помилкову класифікацію. Для оцінки розбиття в цілому використовують середню величину силуету по всіх об'єктах. Оптимальна кількість кластерів за методом силуетів – це таке  $k$ , при якому спостерігається максимум середнього силуету. На практиці будується графік середнього силуету залежно від  $k$ , і вибирається  $k$  з найбільшим значенням. Цей підхід має перевагу у вигляді чіткішого критерію, адже замість візуального пошуку згину отримується числова міра, яку можна максимізувати. Силует має кілька переваг, зокрема враховує одночасно компактність кластерів (низькі внутрішні відстані  $a$ ) та їх роздільність (високі міжкластерні відстані  $b$ ), при цьому не залежить від конкретного алгоритму кластеризації, що робить його придатним як до  $k$ -means, так і до ієрархічної кластеризації та інших методів. В умовах логістики коефіцієнт силуету дає уявлення, наскільки чітко групуються товари за своїми профілями попиту. Вищі значення силуету означають, що утворені групи SKU мають справді відмінні профілі, наприклад, один кластер об'єднує товари з рівномірним стабільним попитом, інший містить сезонні товари, а ще один включає позиції з нерегулярними сплесками продажів. Низький або близький до нуля силует свідчить про те, що кластери накладаються і чіткої сегментації не досягнуто. На практиці це може свідчити про те, що багато товарів мають схожі змішані патерни попиту, а поділ на групи є радше умовним. Так, у дослідженні складу електронної комерції товари виявилися досить однорідними за динамікою попиту, через що середній силует в оптимальному розбитті (13 кластерів) становив лише 0,22 [68]. Попри таке невисоке значення, це кластерне розбиття було визнано корисним, оскільки забезпечило виокремлення групи товарів з високою швидкістю обігу для оптимізації зберігання.



Силует добре підходить для оцінки сферичних кластерів у метриці відстаней, хоча його недоліком є те, що за складної форми кластерів або наявності шумових даних він може давати низькі значення навіть за правильної сегментації. Також обчислення силуету при великій вибірці може бути трудомістким, оскільки вимагає порівняння всіх пар об'єктів між кластерами, проте сучасні реалізації та використання прискорювачів, таких як графічні процесори, частково розв'язують цю проблему. У логістиці метод силуету цінний тим, що забезпечує незалежний критерій для вибору  $k$ , оскільки менеджер складу може зіставити результати аналізу за методом «лікоть» із значеннями силуету і прийняти рішення, спираючись на обидва джерела інформації.

**Гар-статистика.** Ще одним підходом для визначення  $k$  є метод гар-статистики, запропонований в роботі [69]. Ідея гар-методу – оцінити, наскільки отримане кластерне розбиття відрізняється від випадкового розподілу точок. Іншими словами, порівнюється фактична внутрішньокластерна варіація даних із її очікуваним значенням для однорідно випадкового розподілу точок у просторі. Величина  $\text{Gap}(k)$  визначається як різниця між логарифмом фактичного WCSS для  $k$  кластерів та середнім значенням логарифма WCSS, обчисленим для випадкових зразків тієї ж розмірності. Інтуїтивно, якщо дані мають виражену кластерну структуру, то при тому ж  $k$  фактичний WCSS буде значно меншим, ніж для випадкових даних, і  $\text{Gap}$  буде великим. Оптимальне число кластерів обирають як найменше  $k$ , при якому  $\text{Gap}(k)$  близьке (в межах однієї стандартної помилки) до  $\text{Gap}(k+1)$ . Метод гар-статистики має статистично строгі підґрунтя і часто виявляється ефективним там, де «лікоть» нечіткий. Його перевага полягає в об'єктивності, адже фактично виконується статистична перевірка на наявність структури в даних шляхом порівняння з випадковим розподілом. Недоліком методу є значні обчислювальні витрати, оскільки для оцінки потрібно згенерувати велику кількість випадкових наборів даних (зазвичай від 100 до 500), кластеризувати кожен із них і обчислити WCSS для подальшого

порівняння. У логістичних задачах гар-метод застосовується рідше через складність, але він може бути корисним, коли потрібно строго обґрунтувати  $k$ . Наприклад, якщо методи «лікоть» та силует дають різні рекомендації, що можливо за незначної різниці в якості кількох варіантів розбиття, гар-статистика може слугувати незалежним орієнтиром і показати, чи не є вибране розбиття надмірно складним порівняно з випадковою моделлю. Умовно кажучи, якщо значення гар-метрики досягає максимуму при  $k = 4$ , це свідчить про те, що чотирикластерна структура є значущою і, ймовірно, відображає реальні патерни попиту.

**Індекс Девіса–Болдуїна.** Серед класичних внутрішніх метрик якості кластеризації варто згадати індекс Девіса–Болдуїна ( $DB$ ). Цей індекс базується на співвідношенні внутрішньокластерної та міжкластерної відстаней. Розрахунок передбачає, що для кожного кластера  $i$  визначається інший кластер  $j$  для якого величина  $\frac{\Delta_i + \Delta_j}{d(i,j)}$  є максимальною. У наведеній формулі параметр  $\Delta$  позначає середню відстань об'єктів до центру власного кластера, а  $d(i,j)$  – відстань між центроїдами порівнюваних кластерів. Індекс  $DB$  є середнім значенням цієї величини по всіх кластерах. Чим менше значення  $DB$ , тим кращим вважається кластерне розбиття, оскільки низькі значення свідчать про компактність кожного кластера (невелике  $\Delta$ ) та його значну віддаленість від найближчого сусіда (високе  $d(i,j)$ ). На відміну від силуету, індекс Девіса–Болдуїна не обмежений зверху 1 і може набувати будь-яких невід'ємних значень. Для вибору оптимального  $k$  обчислюють  $DB$  для різних  $k$  і обирають той, при якому індекс мінімізується. До переваг індексу  $DB$  належать його простота, оскільки формула ґрунтується лише на середніх відстанях і центроїдах, а також висока швидкість розрахунку, що спрощує застосування до великих вибірок [62, 66]. До того ж, на відміну від силуету,  $DB$ -індекс менш залежний від форми кластерів і здебільшого реагує тільки на їх розмір і віддаленість. У випадках, коли кластери товарів можуть суттєво відрізнятися за кількістю SKU та щільністю, індекс  $DB$  може бути корисним інструментом

для оцінки якості кластеризації. Наприклад, якщо при значенні  $k = 5$  індекс набагато менший, ніж при  $k = 4$  або  $k = 6$ , це свідчить про те, що п'ятикластерна структура забезпечує кращу компресію даних, тобто меншу розпорошеність товарів у межах кластерів і кращу їх відокремленість. Проте, як і інші внутрішні метрики, DB-індекс не враховує бізнес-критеріїв. Через це може надаватися перевага варіантам, які покращують лише математичні показники, але не підвищують реальну ефективність роботи складу.

**Логістичні показники ефективності кластеризації.** Вибір  $k$  не можна здійснювати у відриві від практичних цілей кластеризації. У задачах зберігання та комплектації замовлень якості кластерного розбиття бажано оцінювати не лише внутрішніми метриками, а й за допомогою логістичних **ключових показників ефективності** (*Key Performance Indicators, KPI*). До них належать середній час доступу до товару, довжина маршруту комплектувальника та кількість переміщень товарів. Слід забезпечити баланс у завантаженні зон, щоб уникнути надмірного навантаження на окремі ділянки складу. Правильно вибране кластерне розбиття може суттєво поліпшити ці показники. Зокрема, якщо товари згруповані за інтенсивністю попиту, то позиції з високим оборотом доцільно розміщувати у зонах першочергового доступу, розташованих неподалік пунктів видачі. Це скорочує середню відстань, яку проходять комплектувальники, і зменшує час виконання замовлень [49, 50, 70].

Результати моделювання та прикладних досліджень показують, що кластеризація SKU за патернами попиту та відповідне зонування складу здатні знизити сумарний шлях при збиранні замовлень, оптимізувати використання простору й зменшити навантаження на окремі ділянки. У термінах метрик це може означати суттєве скорочення середньої довжини маршруту переміщення персоналу та вирівнювання навантаження між зонами, оскільки кластери товарів із високою частотою попиту розміщуються у просторах і зручно розташованих ділянках складу, що запобігає скупченню в проходах. Іншим важливим показником є кількість переміщень товару. Якщо кластеризація

враховує стабільність попиту, то товари зі стабільно низьким попитом можуть зберігатися на віддалених стелажах без частих перестановок, тоді як продукцію з динамічним або змінним попитом варто розміщувати ближче та забезпечувати їй гнучкіші умови переміщення [49, 70].

Таким чином, оптимальне значення  $k$  має враховувати не лише високі оцінки за внутрішніми метриками (WCSS, силует, індекс DB тощо), а й орієнтуватися на покращення реальних бізнес-показників складу.

**Комбіновані підходи до вибору  $k$ .** На практиці доцільно поєднувати результати внутрішніх метрик із аналізом зовнішніх (бізнесових) KPI. Спочатку визначаються кандидати для  $k$  за допомогою описаних вище методів, наприклад, метод ліктя вказує на 3–4 кластери, максимум значення силуету – на 4–5, а мінімум індексу Девіса–Болдуїна – на 4 [66, 71, 72]. Далі для кожного варіанта кластерного розбиття оцінюють його вплив на логістичні показники, моделюючи розміщення товарів за кластерами та розраховуючи очікуваний час відбору, загальну відстань переміщення комплектувальників, кількість зон та інші параметри. Наприклад, якщо приріст середнього силуету при переході від 4 до 5 кластерів незначний, зате модель показує істотне зменшення довжини маршруту комплектувальника, можна обрати 5 кластерів як компромісний варіант. В інших випадках, навпаки, може з'ясуватися, що розбиття на більшу кількість кластерів лише незначно покращує KPI або навіть погіршує їх через ускладнення процесів, тому доцільніше обрати менше значення  $k$ . Подібний підхід реалізовано дослідниками у вигляді дворівневого критерію, який спершу передбачає оптимізацію внутрішньої метрики для звуження діапазону значень  $k$ . Наступним етапом є точне визначення оптимального  $k$  у симуляційній моделі складу на основі мінімального часу комплектування або іншого важливого бізнес-показника [50, 70]. У результаті комбінований підхід гарантує, що вибране кластерне розбиття є не лише математично обґрунтованим, але й справді оптимальним для цілей зберігання, оскільки забезпечує швидший доступ до товарів, збалансоване навантаження на зони та загальне підвищення продуктивності складських операцій.

Таким чином, вибір оптимальної кількості кластерів є багатоаспектним завданням, що поєднує аналіз структури даних та врахування логістичних вимог. Застосування метрик на кшталт elbow, силуету, гар-статистики та індексу Девіса–Болдуїна допомагає кількісно обґрунтувати вибір  $k$ , мінімізуючи суб'єктивність. Однак остаточне рішення слід ухвалювати з урахуванням практичних наслідків, адже оптимальне кластерне розбиття має бути не лише статистично стійким, а й забезпечувати відчутне підвищення швидкості та ефективності роботи складу.

## **Висновки до розділу 2**

У даному розділі розглянуто методи побудови та кластеризації часових рядів попиту на товари з урахуванням сезонних, поведінкових та зовнішніх чинників. Розроблена модель формування часових рядів базується на агрегуванні історичних даних із різнорідних джерел, зокрема ERP, WMS, CRM систем та інших інформаційних підсистем підприємства. Детально описано як має виконуватись очищення аномалій, заповнення пропусків, а також врахування впливів свят, маркетингових активностей і зовнішнього середовища. Проведено аналіз вибору частоти рядів і джерел даних, що безпосередньо впливають на якість подальшої кластеризації.

Досліджено підходи до кластеризації часових рядів із використанням різних метрик відстані, таких як Euclidean, DTW та SBD, а також алгоритмів кластеризації, зокрема АНС,  $k$ -shape,  $k$ -medoids і SOM. Виконано порівняння ефективності зазначених методів за критеріями інтерпретованості, стійкості до часових зсувів та масштабованості. Розглянуто та доведено доцільність застосування DTW та SBD для групування товарів за формою попиту, а також SOM для візуалізації кластерної структури.

Особливу увагу приділено визначенню оптимальної кількості кластерів із використанням методу ліктя, коефіцієнта силуету та гар-статистики. Запропоновано комбіновану стратегію кластеризації, адаптовану до логістичних обмежень, яка становить інноваційний підхід до сегментації товарів на основі часових рядів попиту.

## РОЗДІЛ 3. РОЗРОБКА ТА ПЕРЕВІРКА МОДЕЛІ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ТОВАРІВ НА СКЛАДІ

### 3.1 Формалізація критерію оптимізації з урахуванням довжин маршрутів

Проблема оптимального розміщення товарів на складі (SLAP) полягає у визначенні таких місць зберігання для кожного товару, щоб мінімізувати витрати на його оброблення, зокрема відстань і час, які витрачає комплектувальник на збирання замовлень. У класичній постановці SLAP формулюється запитання: «Яке найбільш підходяще місце зберігання для кожного товару, щоб мінімізувати довжину маршруту під час його відбору?» [22, 73]. Вхідними даними задачі є характеристики попиту на товари (частота та обсяг замовлень для кожного SKU), а вихідними – оптимальні локації їх розміщення. Загальний випадок задачі належить до класу NP-складних комбінаторних оптимізацій. Такі задачі характеризуються відсутністю відомих поліноміальних алгоритмів пошуку оптимального розв'язку. Підтвердити коректність уже знайденого результату можна за обмежений час, оскільки складність алгоритму описується степеневою функцією від обсягу вхідних даних. Натомість за умови експоненційної залежності трудомісткості обчислень зростає значно швидше. У контексті SLAP це означає, що потрібно знайти найкращий розподіл  $N$  товарів по  $M$  комірках серед величезної кількості можливих комбінацій.

**Критерій оптимізації.** Формально метою є мінімізація сумарної довжини маршрутів комплектування усіх замовлень за певний період. Нехай  $O = \{O_1, O_2, \dots, O_K\}$  – множина замовлень за період, а  $P = \{p_1, p_2, \dots, p_N\}$  – множина товарів (SKU). Будь-яке замовлення  $O_k$  є підмножиною товарів  $P$  (з визначеними кількостями кожного), які необхідно відібрати. План розміщення – це відображення  $\chi : P \rightarrow L$  всіх товарів на множину доступних місць зберігання  $L = \{l_1, \dots, l_M\}$  (комірок на складі). Введемо функцію  $D(O_k, \chi)$  –

довжина шляху, який долає комплектувальник, збираючи замовлення  $O_k$  при розміщенні товарів згідно з планом  $\chi$ . Тоді цільова функція сформульована як:

$$F(\chi) = \sum_{k=1}^K D(O_k, \chi), \quad (3.1)$$

де  $F(\chi)$  – сумарна дистанція або час на виконання всіх замовлень за планом  $\chi$ . Функція  $D(O_k, \chi)$  залежить як від розміщення товарів  $\chi$ , так і від обраної стратегії переміщення комплектувальника по складу (маршрутизації). Оптимізація  $F(\chi)$  здійснюється з урахуванням таких обмежень на план розміщення, зокрема принципу «один товар – одна комірка». Кожен товар  $p_i$  може бути закріплений тільки за однією коміркою складу (принцип dedicated storage), тобто  $\chi(p_i)$  визначається однозначно і різні товари не ділять одну комірку [74]. Обмеження місткості передбачає, що кожна комірка вміщує не більше визначеної кількості одиниць одного SKU, наприклад, не більше 100 штук. Цей ліміт визначає максимальний запас товару, що може бути одночасно розміщений в одній локації. Припускаємо, що цього достатньо для покриття типового добового попиту на кожен обраний товар; у разі перевищення надлишок зберігався б в іншій, сусідній комірці або подібному місці зберігання. Цей нюанс не моделюється, оскільки основна увага приділяється маршрутам комплектування [75]. Однорівневе зберігання означає, що розглядається одноярусний склад із підлоговим розміщенням товарів без багатоярусних стелажів. Вертикальний фактор підйому чи спуску не враховується, що спрощує моделювання, оскільки всі комірки знаходяться на одному рівні доступу для комплектувальника [76]. Статичне розміщення означає, що план  $\chi$  залишається фіксованим протягом усього періоду моделювання, а динамічне переміщення товарів не здійснюється. Це відповідає практиці стабільного розміщення принаймні на горизонті аналізу, коли товари не переміщують після початкового розкладення [73].

Таким чином, критерій мінімізації сумарного шляху комплектувальників безпосередньо впливає на зниження логістичних витрат. Це стосується, зокрема, економії робочого часу персоналу та зменшення експлуатаційних витрат техніки. Відомо, що на пересування по складу під час комплектування припадає до 50 % загального часу обробки замовлень [8]. Отже, скорочення маршрутів має суттєвий економічний ефект.

**Моделювання маршрутів комплектування.** Обчислення  $D(O_k, \chi)$  для конкретного замовлення здійснюється на підставі певних правил маршрутизації комплектувальника. Задача побудови оптимального маршруту для збору замовлення сама по собі є складною і в ідеалі зводиться до задачі комівояжера по точках відбору всередині складу. Однак на практиці використовуються різні евристичні політики маршрутизації [77], що дають наближено оптимальні або принаймні зручні для реалізації шляхи обходу складу. До найпоширеніших належать **S-подібний** (*S-shape*), коли комплектувальник повністю проходить кожен ряд в якому є хоча б один потрібний товар, формуючи маршрут у вигляді літери S (рис. 3.1, а). Після останнього потрібного проходу він повертається до стартової точки (зони видачі) через фронтальний коридор. Ряди без відбору оминаються і працівник у них не заходить [78, 79]. Така стратегія зручна тим, що не потребує розворотів всередині проходу, оскільки зайшовши в ряд, працівник проходить його весь до кінця. S-подібний маршрут часто використовується на складах із довгими проходами та значною кількістю позицій, оскільки спрощує шлях і особливо ефективний при високій щільності відбору [80]. **Маршрут із серединою проходу** (*mid-point*) передбачає поділ кожного проходу на дві умовні половини, де передня розташована ближче до входу, а дальня – на більшій відстані від нього (рис. 3.1, б). Якщо всі потрібні товари в проході знаходяться в передній половині, комплектувальник заходить з переднього боку і, дійшовши до найвіддаленішої потрібної точки (не далі середини проходу), повертається назад тим же шляхом. Якщо ж у проході є товари в дальній половині, комплектувальник заходить з протилежного (тильного) боку



або проходить весь прохід наскрізь до дальнього коридору. Фактично, перший і останній задіяні проходи можуть проходитися повністю, а проміжні – лише до середньої лінії. Спрощено, склад має два коридори фронтальний і тильний; відбори у передній частині проходів здійснюються з боку фронтального коридору, а в дальній – з боку тильного. Стратегія *mid-point* скорочує зайві відстані у проходах з малою кількістю відборів. Вона є простішою у реалізації, проте за великої кількості позицій поступається складнішим алгоритмам, таким як *largest gap* або *S-shape*. Водночас її застосування є доцільним у випадках, коли середня кількість позицій на один прохід невелика [80].

**Маршрут найбільшого розриву** (*largest gap*) – ще одна популярна евристика маршрутизації комплектувальника, що фактично розвиває ідею політики повернення (рис. 3.1, в). У межах проходу визначається найбільший інтервал без необхідних відборів (так званий «найбільший розрив»), який і пропонується оминати. Іншими словами, комплектувальник заходить у прохід з того боку, де знаходиться ближча група потрібних товарів, доходить до кінця цієї групи та повертається назад, не проходячи далі через найбільший порожній відрізок проходу. Якщо ж частина товарів знаходиться і по інший бік найбільшого розриву, то їхній відбір здійснюється окремо – зайшовши з протилежного боку проходу. Така політика дозволяє мінімізувати зайві переміщення порівняно зі стратегією *mid-point*, де межа (середина проходу) фіксована наперед [81]. Маршрут найбільшого розриву часто демонструє меншу загальну довжину ніж *S*-подібний або маршрут з поверненням, особливо коли пункти відбору в проходах зосереджені групами біля різних кінців. Наприклад, на рис. 3.1, в зображено ситуацію, коли товари у проході згруповані поблизу протилежних виходів; *largest gap* стратегія передбачає окремий захід з кожного боку, уникаючи проходження середньої незадіяної частини проходу.

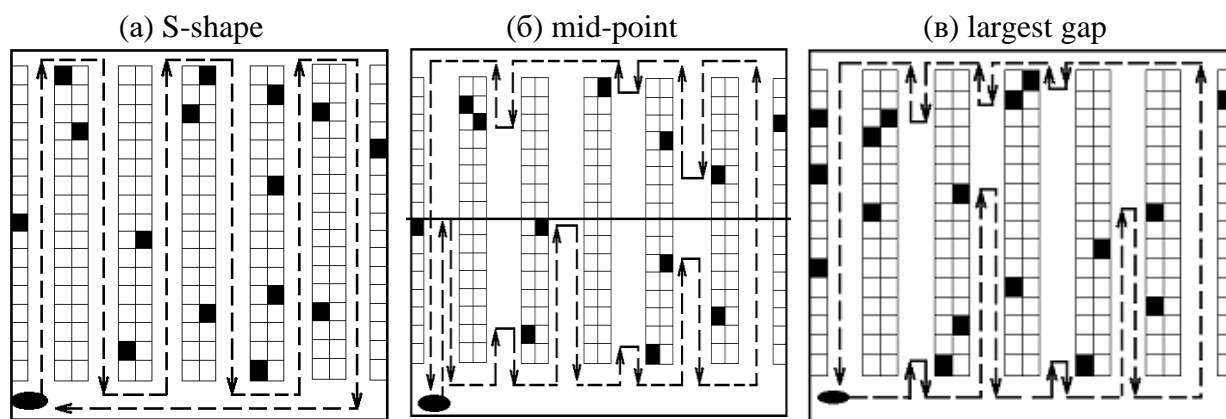


Рисунок 3.1 – Схематичне зображення поширених маршрутів комплектування: (а) S-подібний; (б) маршрут із серединою проходу; (в) маршрут найбільшого розриву

Видно, що S-подібний метод передбачає повне проходження кожного потрібного ряду, тоді як інші стратегії дозволяють раніше повертати назад або заходити з протилежного кінця, скорочуючи тим самим зайві переміщення.

**Маршрут з поверненням** (*return routing*) є більш прямолінійною стратегією. Комплектувальник заходить у кожен прохід, що містить необхідний товар, з одного й того ж боку (зазвичай фронтального), рухаючись лише до найбільш віддаленої потрібної позиції, після чого розвертається і повертається назад до головного коридору [8, 78]. Він не переходить у тильний коридор через прохід, а натомість заходить у наступні потрібні ряди знову з фронтального боку, як це показано на рис. 3.2, а. Такий «петльовий» маршрут простий у реалізації й зручний для працівника, особливо коли пункти відбору переважно розташовані у передній частині стелажних рядів. Стратегія повернення веде до дубльованих переміщень, якщо багато товарів розміщені глибоко в проходах. Оскільки комплектувальник має долати шлях до дальнього пункту двічі, цей підхід програє іншим варіантам [78, 82].

**Комбінований маршрут** (*composite routing*) намагається об'єднати переваги різних стратегій, адаптуючи маршрут під розташування товарів. По суті, для кожного проходу комплектувальника застосовується або повне проходження, або повернення залежно від того, який варіант дає меншу

додаткову відстань [8, 79]. Наприклад, комплектувальник може здійснювати повернення у тих рядах, де всі необхідні позиції знаходяться ближче до входу, і водночас проходити наскрізь ряди з товарами в глибині, щоб вийти у тильний коридор і скоротити шлях до наступних точок [79]. В результаті комбінований маршрут може наблизитися до оптимального за довжиною, оскільки враховує конфігурацію кожного замовлення. Зокрема, така стратегія демонструє стабільно добрі результати за різних конфігурацій складу та списків відбору, хоча і є дещо складнішою у реалізації [8, 82]. На рис. 3.2, б наведено приклад комбінованого маршруту, що включає як «петлі» повернення, так і повні проходи вздовж рядів.

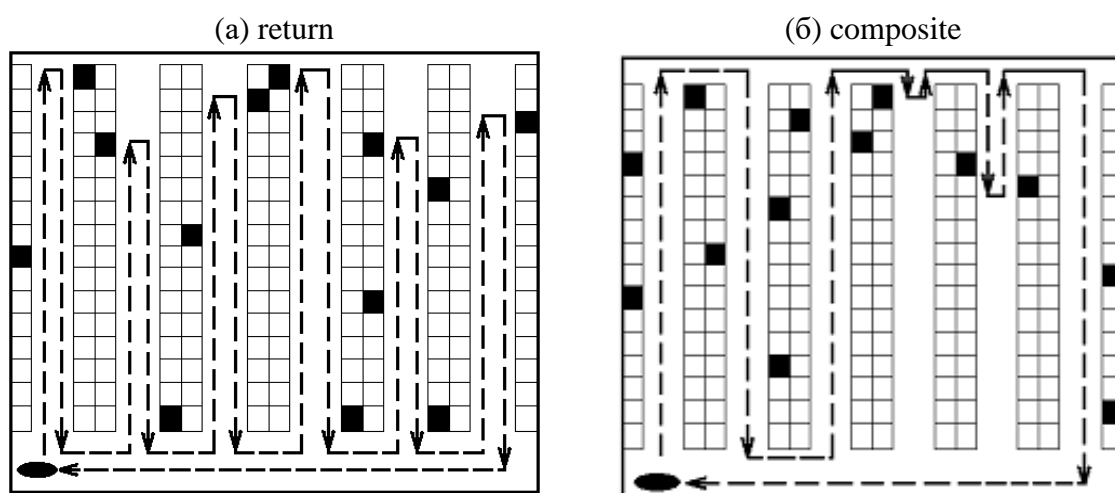


Рисунок 3.2 – Приклади маршрутів комплектування за різними політиками: (а) маршрут з поверненням; (б) комбінований маршрут

Для політики повернення характерно, що працівник після досягнення найдальшої потрібної позиції в проході вертається тим самим шляхом (послідовні петлі заходу в кожен ряд). Натомість комбінована політика дозволяє в деяких рядах пройти наскрізь і вийти у протилежний коридор, обираючи для кожного проходу варіант із найменшою відстанню.

У дослідженні передбачається використання п'яти різних політик маршрутизації, а саме: S-подібного маршруту, маршруту із серединою проходу, маршруту найбільшого розриву, маршруту з поверненням та

комбінованого маршруту. Застосування цих політик під час моделювання дасть змогу оцінити, як різні стратегії руху комплектувальників впливають на загальну довжину маршрутів за однакових умов розміщення товарів. Для кожного замовлення  $O_k$  значення  $D(O_k, \chi)$  визначатиметься як довжина траєкторії комплектувальника, розрахована згідно з правилами однієї з п'яти політик для заданого списку товарів, розташованих відповідно до плану  $\chi$ . Такий підхід дозволяє не лише порівняти ефективність різних стратегій, а й визначити найбільш придатну політику для конкретного типу складу та номенклатури [8, 82].

Важливим аспектом є те, що надмірна орієнтація виключно на мінімізацію відстаней під час комплектування може призвести до ігнорування витрат на іншу важливу операцію, якою є поповнення запасів або **розміщення товарів** (*put-away*). У деяких випадках оптимальне з погляду комплектування розташування товару може вимагати значно більшого шляху під час його первинного переміщення на зберігання. Дослідження у сфері управління складськими операціями свідчать, що недооцінка цього фактора здатна призвести до зростання сукупних експлуатаційних витрат [10, 83]. Тому для забезпечення комплексної ефективності пропонується враховувати як етап комплектування, так і етап розміщення товарів у єдиній оптимізаційній моделі.

Для цього вводимо комбіновану цільову функцію, що мінімізує сумарну довжину маршрутів на складі за обома видами операцій:

$$Z = \alpha \cdot TotalPickingDistance + \beta \cdot TotalPutawayDistance, \quad (3.2)$$

де *TotalPickingDistance* – сумарна довжина маршрутів комплектувальників під час збору всіх замовлень за досліджуваний період. *TotalPutawayDistance* – сумарна довжина маршрутів, які проходять працівники під час розміщення всіх прийнятих товарів на зберігання.  $\alpha$  і  $\beta$  – вагові коефіцієнти, що відображають пріоритетність кожного з етапів операцій та встановлюються підприємством.

Вибір значень  $\alpha$  та  $\beta$  визначає стратегічний баланс між двома компонентами витрат. Наприклад, якщо для підприємства критично важлива швидкість оброблення замовлень і SLA-показники доставки, значення  $\alpha$  може бути встановлено більшим за  $\beta$  ( $\alpha > \beta$ ). У випадках, коли основний обсяг витрат пов'язаний із процесом приймання і розміщення товарів, ваговий коефіцієнт  $\beta$  доцільно збільшити. Значення коефіцієнтів можуть визначатися на основі:

- аналізу історичних даних про тривалість та вартість кожної операції;
- моделювання різних сценаріїв з подальшою оцінкою їх економічного ефекту;
- експертної оцінки керівників складу або логістичного департаменту.

Більш детально модель можна представити у вигляді:

$$Z = \alpha \cdot \sum_{k=1}^K D_{pick}(O_k, \chi) + \beta \cdot \sum_{i=1}^N D_{put}(p_i, \chi), \quad (3.3)$$

де  $D_{pick}(O_k, \chi)$  – відстань для виконання замовлення  $O_k$  при плані розміщення  $\chi$ , а  $D_{put}(p_i, \chi)$  – відстань, необхідна для розміщення товарної позиції  $p_i$  у комірці, закріпленій за цим товаром згідно з планом  $\chi$ .

Запропонований комбінований критерій відображає повний обсяг переміщень на складі й гарантує, що оптимізація не призведе до непропорційного збільшення витрат на одному етапі заради зменшення на іншому [84]. Метою такої оптимізації є мінімізація загальної відстані, яку проходить персонал складу за розглянутий період, із врахуванням усіх ключових складських процесів.

### 3.2 Побудова симуляційного середовища для оцінки розміщення

#### 3.2.1 Часово-орієнтоване призначення місць зберігання (TOASL)

В межах даного дослідження розроблено методику **Часово-орієнтованого Призначення Місць Зберігання** (*Time-Oriented Assignment of Storage Locations, TOASL*) спрямовану на підвищення ефективності складських операцій за рахунок комплексного урахування часової динаміки попиту на товари [5]. На відміну від традиційних підходів, що базуються виключно на середніх значеннях або частотах замовлень, TOASL враховує періодичні та сезонні коливання, а також часові тренди для кожної позиції номенклатури. Це дозволяє визначати розташування товарів у такий спосіб, щоб мінімізувати сумарні витрати на операції як комплектування, так і розміщення.

Реалізація методики TOASL складається з кількох послідовних етапів. На першому етапі здійснюється попередня обробка та нормування даних транзакцій. Далі за допомогою алгоритмів кластеризації часових рядів виявляються ключові групи товарів. На основі отриманих результатів проводиться зонування складу. Завершальним кроком є впорядкування SKU всередині зон за принципом близькості до точки видачі (I/O). Окремим елементом є симуляційне моделювання різних політик маршрутизації та стратегій відбору, що надає інструментарій для порівняння ефективності альтернативних планів розміщення.

Ключовою особливістю TOASL є інтеграція результатів кластеризації з моделями маршрутизації, що уможливорює адаптивність планів розміщення до змін попиту протягом часу. Запропонований підхід дозволяє знайти оптимальне співвідношення між відстанню комплектування та витратами на розміщення товарів. Також дає змогу проводити багатокритеріальну оптимізацію за допомогою вагових коефіцієнтів, які враховують пріоритети підприємства. У межах подальших етапів роботи буде виконано

експериментальне дослідження та перевірку ефективності й практичної корисності методики TOASL за різних сценаріїв роботи складу. Таким чином, розроблений підхід може стати науково обґрунтованою основою для ухвалення рішень щодо організації складського простору, сприяти зменшенню операційних витрат і підвищенню швидкості виконання замовлень. Це, в свою чергу, позитивно вплине на рівень сервісу для кінцевих споживачів.

### 3.2.2 Архітектура методики TOASL та схема експериментів

Архітектура методики TOASL ґрунтується на поетапному перетворенні вихідних даних у науково обґрунтовані плани розміщення товарів з подальшою симуляційною перевіркою їх ефективності. Методика охоплює повний цикл, починаючи від збору та підготовки вхідної інформації, розрахунку середніх відстаней, які необхідно подолати під час розміщення та комплектування товарів, і завершуючи багатофакторною статистичною оцінкою отриманих результатів. Такий підхід забезпечує відтворюваність експериментів, контроль за якістю проміжних результатів і можливість коригування рішень на будь-якому етапі.

Важливим аспектом архітектури є уникнення однобічної орієнтації лише на ефективності комплектування. У багатьох випадках оптимальне для збирання замовлень розташування товарів може вимагати значно довшого шляху під час їх первинного переміщення на зберігання. Недооцінка цього чинника, як підтверджують дослідження у сфері управління складськими операціями, може призвести до зростання сукупних експлуатаційних витрат. Тому TOASL інтегрує етапи комплектування та розміщення в єдину оптимізаційну модель.

Процес реалізації TOASL складається з чотирьох основних послідовних етапів:

1. **Обробка даних** (*Data Preparation*) включає отримання та очищення сирих транзакційних даних від аномалій та пропусків. Також сюди входить відбір

ключових позицій (ABC/Парето) та побудова нормованих часових рядів попиту у масштабі  $[0,1]$ .

2. **Моделювання** (*Modeling*) передбачає використання кількох алгоритмів кластеризації часових рядів, зокрема k-means із DTW-метрикою, SOM та агломеративної ієрархічної кластеризації. Визначення оптимальної кількості кластерів здійснюється за методом ліктя або за дендрограмою, а якість групування перевіряється коефіцієнтом силуету. Після вибору моделі кожен кластер зіставляється із конкретною зоною складу, у межах якої застосовується внутрішньозонава оптимізація розташування SKU.
3. **Симуляція** (*Simulation*) передбачає моделювання конфігурації складу, що включає визначення кількості проходів та розташування точки I/O. На цьому етапі проводиться тестування різних політик маршрутизації, таких як S-shape, mid-point, Largest Gap, Return та Composite, а також перевірка ефективності стратегій відбору Discrete, Batch, Cluster і Zone. У симуляцію інтегрується розрахунок відстаней як для комплектування, так і для розміщення товарів, що дозволяє формувати комбіновані плани з урахуванням обох складських процесів.
4. **Оцінювання** (*Evaluation*) охоплює обчислення метрик ефективності, передусім середньої та сумарної довжини маршрутів для різних сценаріїв і політик розміщення. Застосовується однофакторний **дисперсійний аналіз** (ANOVA). У разі виявлення значущих відмінностей проводяться множинні попарні порівняння за критерієм Tukey HSD із поданням 95 % довірчих інтервалів для різниць середніх.

Така побудова TOASL гарантує, що отримані плани розміщення будуть збалансованими з погляду витрат на обидва ключові процеси, адаптивними до змін у структурі попиту та ефективними за різних конфігурацій складських операцій.

Взаємозв'язки між основними етапами, а також порядок проведення експериментів за методикою TOASL подано на рис. 3.3.



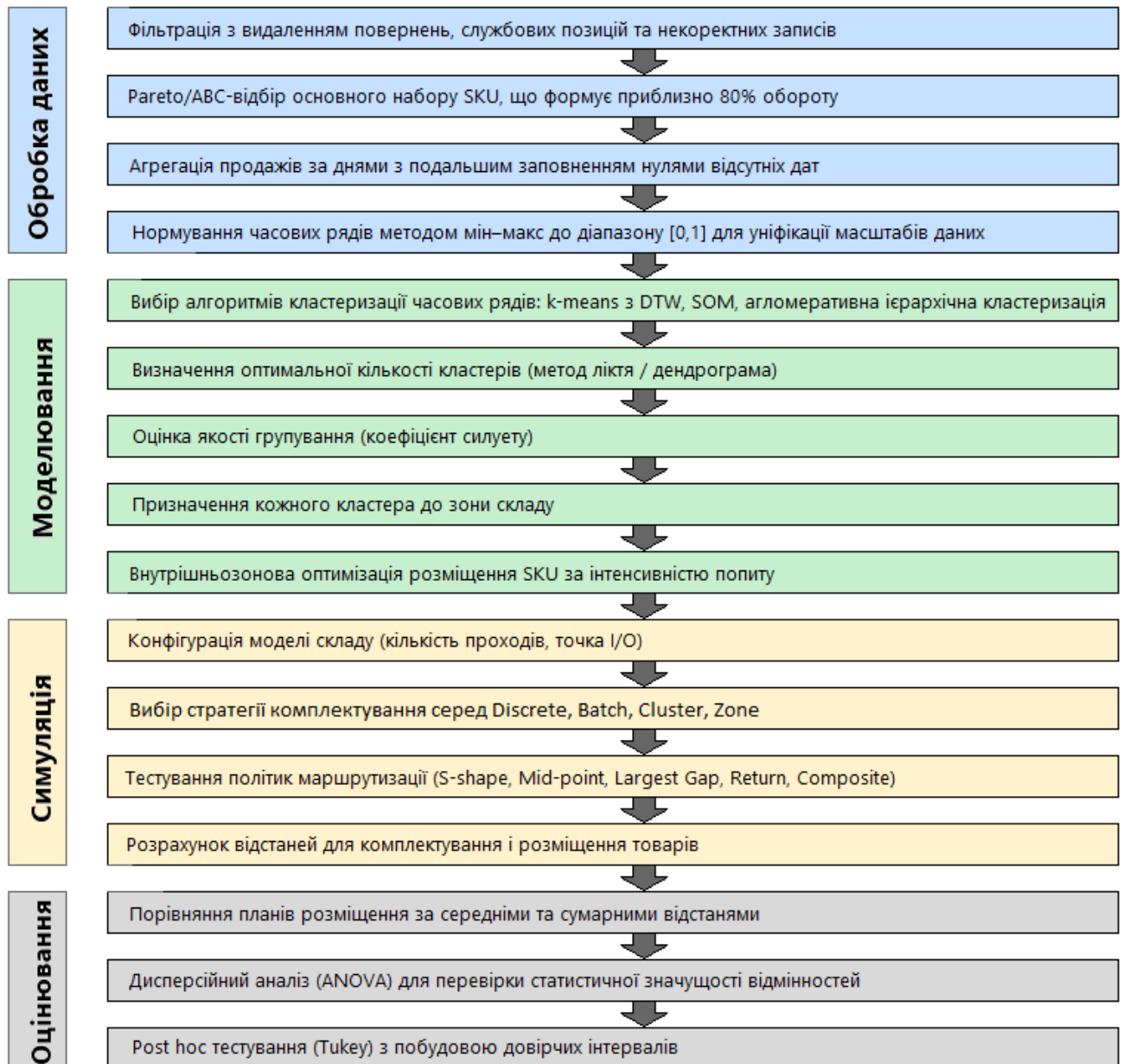


Рисунок 3.3 – Порядок проведення експериментів за методикою TOASL

### 3.3 Опис вихідних даних та їх підготовка

Для моделювання було використано датасет E-commerce dataset – набір реальних транзакцій інтернет-ритейлера, завантажений з платформи Kaggle, що є популярним середовищем для обміну та аналізу відкритих наборів у сфері науки про дані [85]. Він містить детальну історію замовлень за більш ніж 1 рік роботи магазину: з 1 грудня 2010 року по 9 грудня 2011 року. Незважаючи на те що дані були зібрані понад десять років тому, вони залишаються актуальними для дослідження, оскільки відображають типові патерни попиту

в електронній комерції [8, 86]. Це охоплює сезонні піки на кшталт різдвяного періоду, різні частоти та обсяги замовлень по товарах, а також наявність як популярних, так і рідко продаваних позицій. Такий обсяг інформації дозволяє перевірити ефективність нашого підходу кластеризації на реалістичних профілях попиту. Крім того, наявність 541909 записів транзакцій забезпечує статистично надійні результати моделювання, а різноманітність номенклатури, що охоплює 3660 оригінальних товарів, створює можливості для формування груп зі схожою поведінкою.

**Структура даних.** Датасет містить вісім полів, що описують характеристики кожної транзакції: InvoiceNo – номер замовлення, шість цифр, які унікально ідентифікують кожну транзакцію; якщо код починається з літери «С», це означає скасування замовлення. StockCode – код товару, п'ятизначний числовий ідентифікатор, призначений кожному унікальному продукту. Description – назва або опис товару. Quantity – кількість одиниць товару в межах однієї транзакції. InvoiceDate – дата та час оформлення замовлення. UnitPrice – ціна за одиницю товару у фунтах стерлінгів. CustomerID – ідентифікатор покупця, п'ятизначний код, який унікально відповідає кожному клієнту. Country – назва країни у якій проживає покупець.

Кожен запис у датасеті відповідає окремій позиції замовлення, тобто якщо одне замовлення містить три різні товари, воно буде представлено трьома окремими записами з однаковим номером InvoiceNo. Така структура дозволяє точно реконструювати повний набір товарів у кожному замовленні та аналізувати їх у розрізі клієнтів, часу та географії.

	A	B	C	D	E	F	G	H
1	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
2	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2,55	17850	United Kingdom
3	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3,39	17850	United Kingdom
4	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2,75	17850	United Kingdom
5	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3,39	17850	United Kingdom
6	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3,39	17850	United Kingdom
7	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	12/1/2010 8:26	7,65	17850	United Kingdom
8	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	12/1/2010 8:26	4,25	17850	United Kingdom
9	536366	22633	HAND WARMER UNION JACK	6	12/1/2010 8:28	1,85	17850	United Kingdom
10	536366	22632	HAND WARMER RED POLKA DOT	6	12/1/2010 8:28	1,85	17850	United Kingdom
11	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	12/1/2010 8:34	1,69	13047	United Kingdom
12	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	12/1/2010 8:34	2,1	13047	United Kingdom
13	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	12/1/2010 8:34	2,1	13047	United Kingdom
14	536367	22749	FELTCRAFT PRINCESS CHARLOTTE DOLL	8	12/1/2010 8:34	3,75	13047	United Kingdom
15	536367	22310	IVORY KNITTED MUG COSY	6	12/1/2010 8:34	1,65	13047	United Kingdom
16	536367	84969	BOX OF 6 ASSORTED COLOUR TEASPOONS	6	12/1/2010 8:34	4,25	13047	United Kingdom

Рисунок 3.4 – Фрагмент вихідних даних набору E-commerce dataset

**Підготовка даних для аналізу.** Було виконано ряд кроків з очищення та фільтрації, щоб підготувати набір до обробки. Некоректні записи з кількостями товарів  $Quantity \leq 0$  видалили, що дозволило усунути повернення товарів та можливі помилки. Також виключено записи без ідентифікатора клієнта чи без опису товару (поля CustomerID або Description не заповнені), оскільки такі дані неповні і не несуть корисної інформації про попит. Виключення службових позицій здійснювалося для кодів, що не є реальними товарами, а відображають службові операції, зокрема доставку, оплату банківських послуг, нарахування знижок та інші подібні записи. Сформовано список таких кодів до якого увійшли POST, BANK CHARGES, DOT, DOTCOM, AMAZONFEE, CRUK, PADS, DCGSSBOY, DCGSSGIRL, а також однолітерні коди B, D, M, S. Всі рядки зі StockCode, що входять до цього списку, були видалені як нерелевантні для аналізу попиту на товари. Обчислення показників продажів виконувалося після фільтрації шляхом додавання поля TotalSales, яке відображає сумарний дохід за даним рядком та розраховується як добуток Quantity на UnitPrice. Це дозволило оцінити вагомість кожного товару у загальному обсязі продажів [87].

Після очищення та фільтрації даних у наборі залишилося 18405 унікальних замовлень разом із відповідними товарними позиціями, які пройшли перевірку на коректність і повноту. Наступним етапом стало

виділення ключових товарів за принципом Парето з метою зосередження подальшого аналізу на тих позиціях, що формують основну частину обороту.

**Аналіз Pareto-товарів.** Відомо, що в торгівлі часто близько 20 % товарів забезпечують приблизно 80 % обороту, що відповідає принципу ABC-аналізу або правилу Парето. Вирішено зосередити оптимізацію розміщення на товарах з найвищим оборотом, що мають суттєвий вплив на загальну ефективність і належать до категорії А. Відібрано позиції, сукупний виторг від яких становить 80 % від загального обсягу продажів. Спочатку весь асортимент було відсортовано за спаданням показника TotalSales, що відображає сумарний продаж за аналізований період. Обчислено суму продажів у цьому відсортованому списку і визначено порогове значення, що становить 80 % від загального обсягу реалізації. Визначено множину Pareto-товарів, тобто підмножину А, що належить до загальної множини Р, яка охоплює всі позиції від початку списку до тієї, включення якої забезпечує досягнення 80 % від загального виторгу [8, 88].

В результаті до групи А увійшло 784 товарні позиції, що становить 21,42 % від загальної кількості SKU у наборі даних. Таким чином, аналіз даних підтвердив принцип Парето, за яким приблизно 20 % асортименту забезпечили 80 % грошового обороту інтернет-магазину. Ці 784 SKU розглядаються надалі як пріоритетні позиції для оптимального розміщення на складі, оскільки саме для них оптимізація розташування є найбільш критичною. Решту менш популярних товарів, що належать до категорій В і С за ABC, можна зберігати на віддаленіших позиціях або застосовувати до них довільне розміщення, оскільки вони менше впливатимуть на загальний час комплектування. Зосередження на топ-784 товарах дозволяє зменшити розмір задачі, не втрачаючи основного ефекту для оптимізації [88].

Варто зазначити, що серед цих пріоритетних позицій переважають різноманітні подарункові вироби, декор, аксесуари та інші товари, що відображають спеціалізацію інтернет-магазину. Основний асортимент представлений продукцією сегмента «подарунки та товари для дому», тобто

виробами для святкових і повсякденних потреб із виразним акцентом на домашній декор, посуд та сервірування, кухонні й побутові дрібниці, пакування подарунків та невеликі стильні аксесуари.

**Підготовка часових рядів попиту.** Ключовою особливістю підходу є використання часових рядів для кластеризації товарів. Замість звичайних показників типу середньої частоти продажів чи сумарного обсягу, враховується динаміка попиту в часі для кожного товару, тобто коли саме відбувалися його продажі протягом року. Це дозволить групувати товари, наприклад, з яскраво вираженим сезонним характером продажів окремо від товарів зі стабільним рівнем попиту чи одноразовими сплесками [86].

Щоб побудувати часовий ряд для товару, необхідно представити його продажі як функцію часу. Було обрано підхід, за якого кожен окремий продаж (транзакція) подається як точка на часовій осі, а проміжки між продажами заповнюються нульовими значеннями. Зокрема, було виконано такі кроки. Поле `InvoiceDate` переведено до єдиного часового формату, на основі якого визначено числовий показник часу в межах року. Запроваджено метрику `MinutesOfYear`, яка для кожного запису визначає кількість хвилин від початку року до моменту здійснення замовлення. Формула

$$MinutesOfYear = (dayOfYear - 1) \times 24 \times 60 + hour \times 60 + minute \quad (3.4)$$

перетворює дату й час на кількість хвилин від умовного початку року.

Для кожного товару  $p_i \in A$  створено окремий масив (вектор) його продажів. Кожна транзакція товару додавалася як запис пари значень  $t = MinutesOfYear$ ,  $q = Quantity$ . Тобто фіксується момент продажу з точністю до хвилини та кількість проданих одиниць. Якщо протягом року товар продавався  $n_i$  разів, то масив міститиме таку ж кількість точок. Для подальшого аналізу потрібно, щоб усі часові ряди мали однакову довжину (кількість точок), інакше їх не можна коректно порівнювати й використовувати в алгоритмах кластеризації. Тому було виконано вирівнювання довжини шляхом додавання елементів-заповнювачів (`padding`).

Знайдено максимальну кількість транзакцій  $N_{max} = \max_i n_i$  серед усіх товарів  $A$ , що відповідає граничній довжині масиву продажів для найбільш затребуваної товарної одиниці. У результаті аналізу з'ясовано, що найбільш популярним товаром у наборі даних E-commerce dataset є позиція зі StockCode = 85123A, для якої значення  $N_{max}$  становить 2035 транзакцій. Потім до кожного масиву товару  $p_i$ , в якому було менше транзакцій ніж  $N_{max}$ , додано відповідну кількість штучних записів з нульовими значеннями: Quantity = 0 і MinutesOfYear = 0. Ці нульові записи не представляють реальних подій, а слугують заповнювачами, щоб вирівняти розміри масивів. Після цього кожен товар представлено вектором довжини  $N_{max}$ , у якому більшість елементів мають нульові значення, що відповідають відсутності транзакцій, тоді як реальні продажі відображені як ненульові значення кількості у відповідних позиціях послідовності [89]. Для усунення розбіжностей у масштабах кількостей продажів між різними товарами було проведено нормалізацію. Кожен часовий ряд (вектор кількостей для товару) нормовано у діапазон [0, 1] за мін-макс методом:

$$q_{i,j}^1 = (q - q_i^{min}) / (q_i^{max} - q_i^{min}) \quad (3.5)$$

де  $q$  позначає фактичну кількість проданих одиниць товару  $q_{i,j}$  для  $j$  – го продажу товару  $i$ ;  $q_i^{min}, q_i^{max}$  – мінімальна та максимальна кількості для товару  $i$  у його часовому ряді. По суті, для кожного SKU значення кількостей продажів були поділені на його максимум. Це вирівнює вплив товарів з різними масштабами. Наприклад, якщо один товар продавався по 100 одиниць за раз, а інший по одній, то після нормування обидва матимуть пікові значення, близькі до 1, і алгоритм кластеризації зосередить аналіз на формі ряду в часі, а не на абсолютних величинах [86, 90].

	A	B
1	Quantity	MinutesOfYear
2	0,014705882	481466
3	0,014705882	481502
4	0,014705882	481532
5	0,014705882	481611
6	0,014705882	481653
7	0,009803922	481661
8	0	0
9	0	0
10	0	0
11	0	0

Рисунок 3.5 – Приклад перетворених даних для одного з товарів після нормалізації та доповнення відсутніх значень нулями

В результаті маємо матрицю даних розміром  $784 \times 2035$ , де кожен рядок – це нормований часовий ряд продажів конкретного товару з Pareto-списку. Горизонтальна вісь – умовний індекс продажу від 1 до  $N_{max}$  в хронологічному порядку. Така модель відображення динаміку продажів дає змогу застосувати стандартну кластеризацію, оскільки кожна послідовність розглядається як точка у багатовимірному просторі розмірності  $N_{max}$  [89]. Альтернативою було б агрегувати дані, наприклад, по днях або тижнях, утворюючи рівномірні часові ряди; однак обраний метод фіксації всіх реальних транзакцій дозволяє потенційно зберегти більше інформації, зокрема часову структуру всередині дня, якщо вона має значення.

У результаті аналізу було встановлено, що набір даних є актуальним та придатним для дослідження. Отримані часові ряди відображають характерні для електронної комерції цикли. Зокрема, для багатьох товарів спостерігаються виразні піки продажів у листопаді, що пов'язані з передріздвяним сезоном, та спад попиту в літні місяці. Такі закономірності притаманні й сучасним онлайн-продажам, тому кластеризація на основі цих часових рядів формує групи, які легко інтерпретуються, наприклад, як сезонні святкові пропозиції, цілорічні хіти продажів чи позиції для разових акцій і

розпродажів та можуть бути ефективно використані для оптимізації складу і сьогодні.

### 3.4 Кластеризація часових рядів та вибір кількості кластерів

**Робоче середовище та обчислювальні налаштування.** Реалізація алгоритмів виконувалася мовою Python у середовищі Google Colab. Colab – це хмарний сервіс, побудований на базі Jupyter-ноутбуків, у якому код виконується у браузері на виділеній віртуальній машині Google, а в одному файлі одночасно зберігаються осередки коду і їхній вивід (графіки, таблиці, логи) разом із пояснювальним текстом. За потреби додаткові бібліотеки, такі як `numpy`, `pandas`, `scikit-learn` та інші встановлюються безпосередньо в ноутбучі за допомогою `pip`, що забезпечує відтворюваність налаштувань.

У налаштуваннях обчислювального середовища було обрано апаратне прискорення v2-8 TPU. У такій конфігурації зазвичай надається віртуальний процесор класу Intel Xeon (приблизно 2–4 vCPU на частоті близько 2.2–2.5 ГГц), оперативна пам'ять орієнтовно 12–25 ГБ та тимчасове дискове сховище приблизно 70 ГБ. Апаратним прискорювачем слугує Cloud TPU v2-8, що містить вісім тензорних ядер із загальним обсягом 64 ГБ HBM та оптимізацією під операції лінійної алгебри у форматі чисел `bfloat16` [91]. Зазначені параметри можуть дещо варіюватися між окремими сесіями Colab, однак наведені діапазони відображають типову конфігурацію для режиму v2-8 TPU і є достатніми для відтворення представлених експериментів.

Для ясності наведено стислий опис деяких термінів. **TPU** (*Tensor Processing Unit*) – спеціалізований прискорювач Google для машинного навчання. На відміну від універсальних CPU та GPU, TPU апаратно прискорює матрично-векторні обчислення, зокрема множення великих матриць, які домінують у нейронних мережах та інших тензорних задачах. Позначення v2-8 означає конфігурацію другого покоління TPU з вісьмома обчислювальними ядрами, з'єднаними високошвидкісною шиною для спільної роботи. Формат



bfloat16 – це 16-бітне представлення чисел із ширшим діапазоном показника степені порівняно з FP16, яке добре підходить для тренування моделей без помітної втрати точності та з меншими вимогами до пам'яті. **HBM** (*High Bandwidth Memory*) – тип багатошарової оперативної пам'яті з дуже високою пропускною здатністю, що розміщується поруч з обчислювальним кристалом і з'єднується широкою шиною. Завдяки цьому HBM забезпечує значно вищу швидкість доступу до даних при нижчому енергоспоживанні ніж звичайна GDDR чи DDR-пам'ять, що особливо важливо для тензорних операцій великого обсягу. У Cloud TPU v2-8 сумарні 64 ГБ HBM слугують спільним буфером для батчів і ваг моделей, дозволяючи обробляти великі масиви даних з мінімальними затримками.

**Вихідні передумови та обґрунтування підходу.** Вище зазначено, що результати аналізу підтвердили наявність типових для електронної комерції сезонних циклів і піків попиту, що робить кластеризацію часових рядів доцільною для подальшого розміщення товарів на складі [8, 86]. У наведеному описі показано, що часові ряди сформовано для 784 SKU із категорії топових позицій за обсягом продажів та зафіксовано виразні піки у листопаді й спад у літні місяці; такі закономірності добре інтерпретуються у термінах «святкові», «цілорічні» й «акційні» групи позицій.

**Методи кластеризації та метрики.** Розглянемо три підходи до групування часових рядів: геометричні, ієрархічні та нейронні. Геометричні методи базуються на використанні метрики подібності, наприклад, евклідової відстані або DTW та намагаються знайти компактні групи навколо центрів. Ієрархічні методи поступово групують об'єкти, формуючи дерево об'єднань, з якого кількість кластерів визначається відсіканням на потрібній висоті. Нейронні методи, зокрема карти Кохонена (SOM), відображають багатовимірні ряди на двовимірну решітку з подальшим групуванням прототипів [92].

**АНС (агломеративна ієрархічна кластеризація, зв'язок Ward).** Кожний ряд початково є окремим кластером, а на кожному кроці зливаються

дві найближчі групи так, щоб мінімізувати приріст внутрішньокластерної дисперсії. Результатом є дендрограма, яка дозволяє візуально оцінити подібність і вибрати кількість кластерів без перезапусків. Підхід добре інтерпретується й стійкий до ініціалізації, але має підвищену обчислювальну складність на великих наборах і чутливий до вибору відстані та масштабування даних [93].

**K-means (евклідова та DTW-версії).** Алгоритм задає кількість кластерів  $k$ , ітеративно оновлює центроїди як середні значення точок та відносить спостереження до найближчого центроїда за обраною метрикою. У евклідовому варіанті кожен часовий ряд розглядається як вектор сталої довжини; така постановка забезпечує високу швидкість і простоту реалізації, проте є чутливою до зсувів у часі й амплітуди. У варіанті з DTW застосовується еластичне вирівнювання траєкторій, що дає змогу точніше об'єднувати подібні форми піків, але вимагає істотно більших обчислювальних ресурсів і складнішого оновлення центроїдів [94]. До сильних сторін належать явний контроль за  $k$  та зрозуміла геометрична інтерпретація. Серед обмежень варто відзначити залежність від ініціалізації та необхідність попередньо фіксувати  $k$ .

**Self-Organizing Map** є нейромережевим підходом, у межах якого спочатку розміщують набір прототипів на решітці  $4 \times 4$ . Далі під час навчання близькі ряди активують сусідні вузли, що зберігає топологію. Після цього прототипи групуються в підсумкові кластери, найчастіше із застосуванням АНС або  $k$ -means [92]. SOM корисна для нелінійних структур і перехресних сезонних патернів, але потребує налаштування гіперпараметрів і зазвичай є обчислювально ресурсоємною.

### 3.4.1 Оцінювання $k$ і перевірка структури за дендрограмою

Оцінювання кількості кластерів для всіх алгоритмів, які потребують наперед заданого  $k$ , виконувалося за методом ліктя. Сутність підходу полягає

в аналізі залежності WCSS (сума квадратів відхилень об'єктів від центрів своїх кластерів) від числа кластерів  $k$  і пошуку такої точки, після якої додаткове збільшення  $k$  дає лише незначне зменшення WCSS. Щоб формалізувати вибір, застосовано евристику максимального відхилення від опорної прямої [67, 95], яка проводиться між першою та останньою точками кривої. Оптимальним вважається те  $k$  для якого перпендикулярна відстань до цієї прямої найбільша.

На рис. 3.6 подано графік WCSS( $k$ ) для діапазону  $k = 2 \dots 60$ . По осі абсцис відкладено кількість кластерів, по осі ординат – значення WCSS, що зменшується зі зростанням  $k$ . Сині точки та лінія відображають фактичні значення WCSS, помаранчева пунктирна пряма з'єднує крайні точки та слугує опорною лінією для оцінки «ліктя», а червона пунктирна вертикаль позначає знайдене значення  $k$ . Крива спадає майже лінійно, але поблизу  $k \approx 16$  спостерігається локальний злам. Відстань від кривої до опорної прямої є максимальною саме в цій точці. Така поведінка свідчить, що подальше збільшення кількості кластерів лише незначно зменшує WCSS й не виправдовує ускладнення моделі. З огляду на це обрано  $k = 16$  як оптимальне співвідношення між компактністю кластерів і стабільністю розбиття.

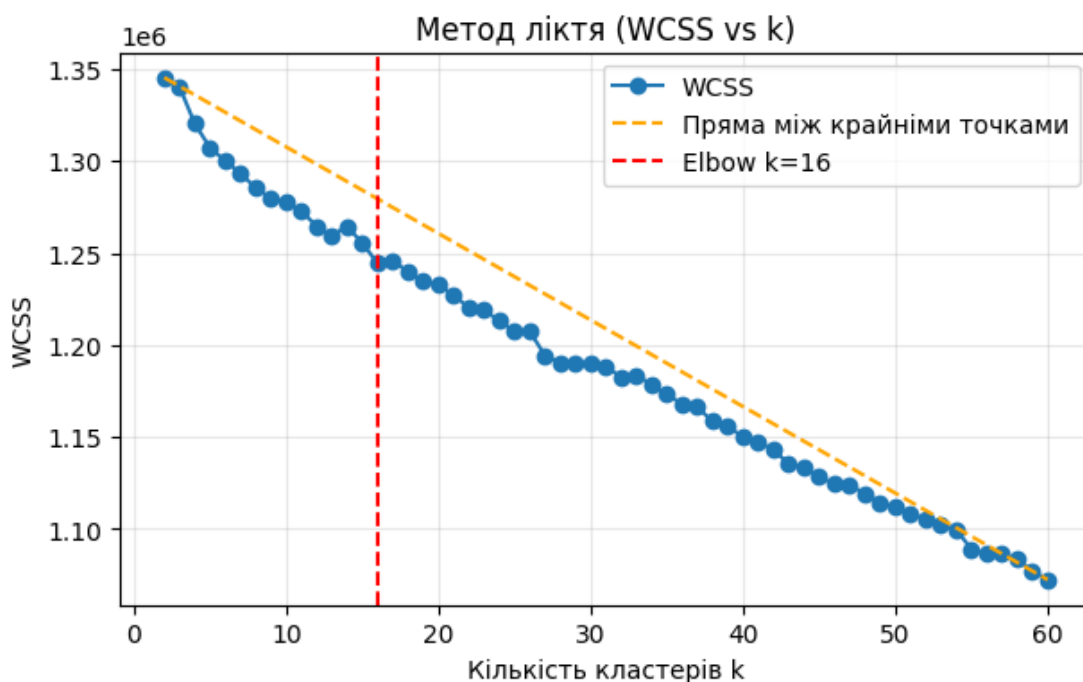


Рисунок 3.6 – Визначення оптимального  $k$  за правилом максимального відхилення

Для додаткової перевірки структури розбиття використано дендрограму агломеративної кластеризації зі зв'язком Ward. Такий графік відображає послідовність злиття і відстані між групами. Листки відповідають окремим часовим рядам, вертикальні відрізки – висоті злиття, а шкала Distance інтерпретується як приріст внутрішньокластерної дисперсії. Чим вищий рівень злиття, тим менш подібні були об'єднані множини.

На рис. 3.7 горизонтальна пунктирна лінія позначає поріг відсікання. Перетин із гілками утворює 16 піддерев. Поріг розташовано нижче перших великих «стрибків» висоти й вище локальних дрібних об'єднань кластерів, тож зберігається внутрішня однорідність і не відбувається змішування різних профілів. Інакше кажучи, поріг встановлено перед ділянкою різкого зростання міжкластерних відстаней, що підтверджує обґрунтованість вибору  $k = 16$  [93].

Дерево також демонструє кілька «макрогілок», що відповідають віддаленим одна від одної групам динаміки попиту, зокрема сезонним і несезонним. У межах кожної макрогілки видно підрозділи різного порядку. Частина підрозділів відображає відмінності у формі та інтенсивності піків, інші – у базовому рівні попиту, а ще інші – у фазових зсувах, коли піки припадають на різні місяці. Обраний поріг фіксує саме ці відмінності середнього масштабу, зберігаючи інтерпретованість кластерів і уникаючи надмірного дроблення.

Для зв'язку Ward «ланцюговий ефект» не спостерігається. Гілки розгалужені, а не витягнуті в довгі послідовності одиничних приєднань. Висоти останніх об'єднань кластерів істотно перевищують рівень порога, що забезпечує достатній запас міжкластерних відстаней і підкреслює стабільність отриманих груп. Отже, дендрограма використовується як інструмент валідації, а не прямого вибору  $k$ . Вона дозволяє виявляти нетипові злиття, бачити природні об'єднання та перевіряти, чи не відбувається надмірне дроблення. Сукупність цих ознак підтримує подальшу роботу саме з 16 кластерами.

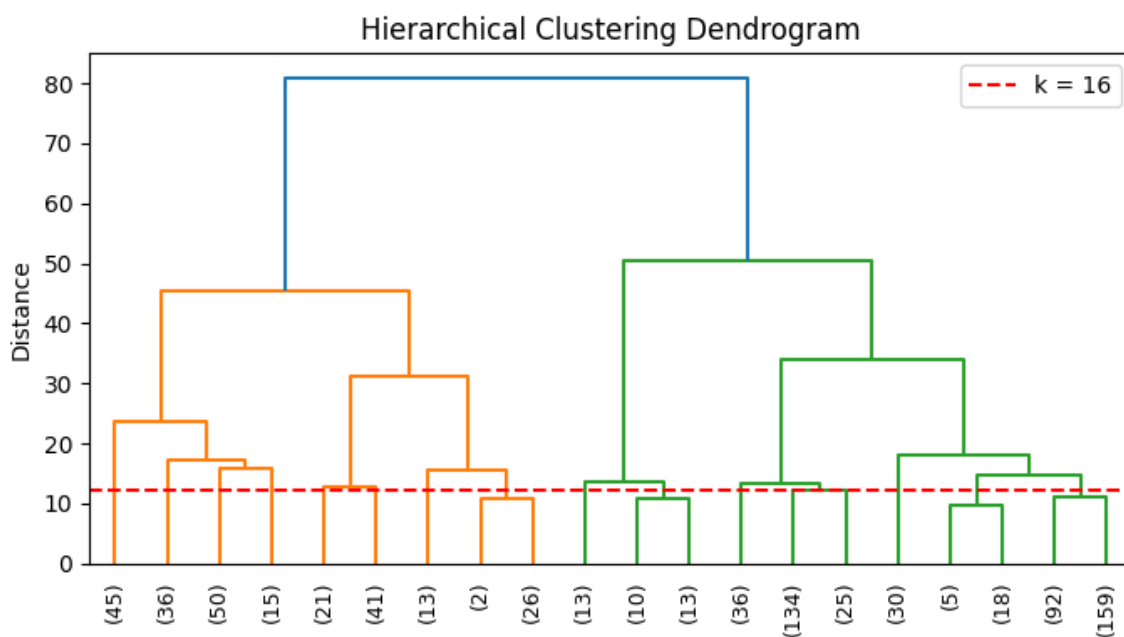


Рисунок 3.7 – Перевірка структури кластерів за дендрограмою з відсіканням на висоті для  $k = 16$

### 3.4.2 Порівняння алгоритмів за узагальненими метриками якості та часу

Щоб обрати базову модель для подальших етапів, усі розглянуті алгоритми оцінювалися за єдиною методикою. Вхідні дані – нормовані часові ряди попиту для 784 SKU з множини ключових товарів. Кількість кластерів фіксовано на рівні  $k = 16$ . Для кожного алгоритму виконувалися багаторазові запуски з однаковими параметрами попередньої обробки, а час виконання вимірювався в одному й тому самому середовищі обчислень. Якість розбиття оцінювали за коефіцієнтом силуету. Для кожного об'єкта порівнювали середні відстані до точок власного кластера та до найближчого сусіднього кластера, після чого обчислювали узагальнене значення. Підсумковий показник належить до інтервалу від  $-1$  до  $1$ , і що він вищий, то чіткіше розділено кластери.

Насамперед слід окреслити розглянуті алгоритми та їх налаштування. По-перше, застосовано агломеративну ієрархічну кластеризацію зі зв'язком Ward. Її мета – мінімізувати приріст внутрішньокластерної дисперсії на

кожному кроці об'єднання, при цьому відстань між рядами обчислювали за евклідовою метрикою. По-друге, використано класичний k-means з евклідовою відстанню, а початкові центроїди ініціалізовано за процедурою k-means++, що відбирає віддалені точки з більшою ймовірністю та підвищує стабільність розв'язку. По-третє, розглянуто варіант k-means із метрикою DTW, у якому перед обчисленням відстаней послідовності еластично вирівнювалися, а оновлення центроїдів виконувалося на вирівняних траєкторіях. До того ж, залучено карту Кохонена з двовимірною решіткою 4×4, а після завершення навчання прототипів їх додатково згруповано у 16 підсумкових кластерів. Задля відтворюваності результатів зафіксовано зерна генераторів випадкових чисел, параметри нормування та межі пошуку гіперпараметрів. Нарешті, під час оцінювання часу враховано повний обсяг обчислень, зокрема кількість ітерацій кожного алгоритму, а саме епохи навчання у SOM і кроки оновлення центроїдів у k-means до досягнення збіжності.

Таблиця 3.1 – Оцінювання алгоритмів кластеризації за коефіцієнтом силуету і часом виконання

Алгоритм	Відстань / критерій	Метод визначення k	Кількість кластерів	Час виконання, с	Коефіцієнт силуету
АНС (Ward)	Евклідова; Ward (мінімізація дисперсії)	«Лікоть» + валідація дендрограмою	16	5	0,36
SOM	DTW на прототипах; решітка 4×4	Розмір решітки 4×4, кількість прототипів = 16	16	45	0,16
K-means (DTW)	DTW-відстань	«Лікоть»	16	210	0,05
K-means	Евклідова відстань	«Лікоть»	16	35	0,03

*Примітка.* Вказано середній час виконання за однакових налаштувань середовища, можливі незначні варіації значень між сесіями.

Нижче наведено детальний аналіз результатів, узагальнених у табл. 3.1, із порівнянням якості кластеризації (коефіцієнт силуету), часових витрат та особливостей налаштувань кожного алгоритму.

1. **АНС (Ward).** Показник силуету дорівнює 0,36 за часу виконання 5 с, що свідчить про найкраще серед порівнюваних методів поєднання якості розбиття та обчислювальної ефективності [5]. Такий результат пояснюється властивістю критерію Ward мінімізувати приріст внутрішньокластерної дисперсії на кожному кроці агломерації, тобто безпосередньо оптимізувати компактність кластерів. Це узгоджується з логікою силуету, який зростає, коли внутрішньокластерні відстані малі, а міжкластерні – великі. Додатковою перевагою є відсутність залежності від випадкової ініціалізації, що забезпечує стабільність розбиття між запусками й підвищує придатність АНС для регулярного оновлення кластерної карти в оперативній експлуатації.
2. **SOM.** Карта Кохонена забезпечує силует 0,16 за 45 с, тобто дає проміжний результат між АНС та варіантами k-means. Механізм збереження топології сприяє формуванню локально узгоджених груп і виокремленню типових форм динаміки попиту, зокрема сезонних піків різної інтенсивності. Водночас етап навчання прототипів і подальше їх групування не забезпечують тієї чіткості меж, яку демонструє АНС на тих самих даних, що й пояснює нижчий силует.
3. **K-means (DTW).** Використання DTW частково усуває проблему зсувів, оскільки послідовності вирівнюються еластичним способом. Силует зростає до 0,05, однак обчислювальне навантаження значно підвищується і час виконання становить 210 с. Це пов'язано з квадратичною складністю DTW за довжиною ряду та з необхідністю виконувати вирівнювання на кожній ітерації й для кожного центроїда. Отже, покращення якості є помірним, тоді як витрати часу і ресурсів істотно зростають.
4. **K-means (евклідова відстань).** Силует дорівнює 0,03 за тривалості виконання 35 с, що вказує на майже відсутню відокремленість груп [96]. Для

часових рядів така картина очікувана, оскільки евклідова метрика чутлива до фазових зсувів і відмінностей амплітуд. Навіть близькі за формою траєкторії, зсунуті у часі, потрапляють до різних кластерів, а траєкторії різної інтенсивності – до одного, якщо вони геометрично близькі у просторі значень. У підсумку припущення k-means про квазісферичні кластери виявляється надто жорстким для досліджуваних часових даних [86, 94].

Узагальнюючи табл. 3.1, можна зробити висновок, що найкращі показники отримано для АНС (Ward), оскільки цей метод формує компактні та добре розділені кластери за прийнятною тривалості виконання і забезпечує стабільність результатів між запусками. Це робить його найбільш придатною базовою моделлю для подальших етапів розміщення товарів та симуляцій комплектування [93].

З огляду на цей вибір подальші дії мають спиратися на кластерну карту отриману за АНС. По-перше, така карта слугуватиме основою для зонування складу. Кластери з найбільшим внеском у продаж доцільно розміщувати у фронтальних зонах, максимально наближених до центральної точки приймання-відвантаження, тоді як сезонні групи варто збирати у компактні блоки з можливістю ротаційного наближення в періоди пікового попиту. Це забезпечує скорочення середньої довжини маршрутів комплектувальника, зменшує кількість перетинів потоків і підвищує пропускну спроможність у критичні інтервали [8]. По-друге, результати кластеризації задають природну шкалу пріоритетів, яку доцільно інтегрувати у комбіновану цільову функцію. Початкові ваги мають відображати очікуваний виграш у комплектуванні від наближення високопродуктивних кластерів до зони відбору і водночас враховувати витрати на розміщення та можливі переміщення запасів. Баланс між скороченням шляху під час відбору й трудомісткістю операцій розміщення запасів варто налаштовувати емпірично, з урахуванням обмежень місткості зон і доступності осередків. По-третє, частина кластерів демонструє помірну міжкластерну відстань, отже всередині зон доцільно застосувати внутрішнє впорядкування. Практично це означає ранжування позицій у межах



кластера за частотою попиту або оборотністю, розміщення найбільш «ходових» товарів у так званих золотих осередках із найкоротшим доступом, а також використання простих евристик на кшталт ABC-упорядкування всередині кожної з груп. Такий крок доповнює виграш від зонування і додатково скорочує середню довжину маршрутів комплектування без зміни глобальної схеми розміщення.

### 3.4.3 Узагальнення результатів кластеризації ( $k = 16$ ) та практичні висновки

Отримане призначення SKU до кластерів за зрізу дендрограми на 16 груп забезпечує повну відповідність кожного товару певному кластеру та дає змогу оцінити структуру асортименту з позицій місткості груп і їхнього внеску в попит. Загалом 784 SKU розподілено на 16 кластерів, при цьому медіанний розмір кластера становить 29 позицій. Такий рівень деталізації є достатнім для подальшого зонування. З одного боку, кластери не надто дрібні, щоб втрачати керованість, а з іншого – зберігають внутрішню однорідність, необхідну для інтерпретації та практичних рішень щодо розміщення.

Розподіл за **обсягом продажів** (*TotalSales*) демонструє чітку концентрацію товарообігу в кількох домінантних групах. Три кластери з найбільшим оборотом № 12, № 11 і № 16 акумулюють близько 59,9 % сукупного обороту сегмента топових позицій. Зокрема, кластер № 12 із 134 SKU відповідає за 24,6 % продажів, кластер № 11 із 36 SKU – за 17,8 %, а кластер № 16 із 251 SKU – за 17,5 %. Водночас серед кластерів вирізняються групи з мінімальним внеском, зокрема № 7 та № 4 кожна з яких формує менше ніж 1 % обороту. Вказана асиметрія свідчить про наявність ключових кластерів, що формують основний потік замовлень, і другорядних груп із локальним або епізодичним значенням.

Змістовна інтерпретація кластерів узгоджується з очікуваною поведінкою попиту. Кластер № 11 містить виразні «хіти» широкого попиту,

зокрема REGENCY CAKESTAND 3 TIER і WHITE HANGING HEART T-LIGHT HOLDER, що є показовим для цілорічних бестселерів. Кластер № 12 зосереджує товари для дому та повсякденні аксесуари, як-от HEART OF WICKER, DOORMAT UNION FLAG/NEW ENGLAND і металеві таблички, тобто типові масові позиції зі стабільним попитом. У кластері № 13 поєднуються змішані сезонні мотиви, зразки на кшталт PAPER CHAIN KIT 50's CHRISTMAS та HOT WATER BOTTLE, що вказує на зимово-святкову динаміку. Кластер № 16, навпаки, агрегує товари для літніх або зовнішніх активностей, такі як WOODEN CROQUET/ROUNDERS SET, PICNIC BAG та UNION JACK BUNTING, і демонструє сезонність, протилежну до новорічної. Такі приклади підкріплюють тезу про те, що кластери фіксують не лише агреговані обсяги, а й подібність форм попиту придатну для логістичної інтерпретації [86].

Аналіз замовлень підтверджує провідну роль найбільших кластерів. Групи № 12, № 11 та № 16 мають найширше представлення в унікальних замовленнях, відповідно 14057, 12051 та 13500 випадків, якщо рахувати наявність у замовленні хоча б однієї позиції з відповідного кластера. За рахунок такої представленості ці кластери формують основний «потік» під час комплектування, а відтак мають розглядатися як пріоритетні кандидати для центральних зон зберігання та найкоротших маршрутів доступу до точки I/O. Натомість кластери з невеликим оборотом, а саме: № 4, № 7, № 9, № 10 та № 15, доцільно розміщувати у віддаленіших осередках або на полицях із нижчим пріоритетом відбору, не перевантажуючи «фронтальні» ділянки.

Практичні наслідки для зонування впливають безпосередньо з цієї структури. Доцільним є призначення кожному кластеру окремої зони або підзони, що зменшує перетини маршрутів між різними збирачами замовлень та полегшує управління потоками. Кластери з найбільшими продажами варто розташовувати ближче до точки I/O, забезпечуючи мінімальні відстані під час відбору найбільш затребуваних позицій. Сезонні групи доцільно консолідувати у компактних секціях, які можна ротаційно наближати в

періоди пікового попиту, мінімізуючи взаємне перетинання маршрутів і перевантаження проходів. Така організація поєднує стратегічне зонування з операційними перевагами, що прямо впливають із виявлених кластерних закономірностей.

### 3.5 Конфігурація складу і параметри симуляції

Окреслимо геометрію складу та правила обчислення пройдених відстаней. Розглядається умовний одноблоковий склад на 784 комірки, що відповідає кількості відібраних SKU, а конфігурація реалізована як 14 паралельних поздовжніх проходів по 56 комірок у кожному з двосторонніми стелажми по 28 позицій на бік. Ряди розташовані паралельно один одному, утворюючи єдиний функціональний блок без внутрішніх поперечних переходів. На фронтальному торці розміщено головний коридор із точкою I/O, яка виконує роль старту та фінішу маршрутів комплектувальника. На протилежному торці передбачено тильний коридор, що забезпечує альтернативний вхід і вихід у кожний прохід. Відсутність внутрішніх перемичок означає, що всі переміщення відбуваються лише вздовж проходів і двох торцевих коридорів, тобто за ортогональною (мангеттенською) метрикою. Під цим розуміємо, що траєкторія руху складається виключно з відрізків, паралельних осям проходів і коридорів без будь-яких діагональних «скорочень». Параметри конфігурації та прийняті позначення наведено на рис. 3.8. Крок уздовж проходу  $v = 1$  м, довжина проходу  $h = 28$  м, відстань між входами сусідніх проходів  $w = 1,5$  м; параметр  $m = 2$  м – це поперечний крок між сусідніми проходами, що конструктивно дорівнює сумарній ширині двох стелажів. Така постановка однозначно задає адресацію комірок і слугує базою для подальших розрахунків довжин маршрутів як під час розміщення товарів, так і при відборі замовлень.

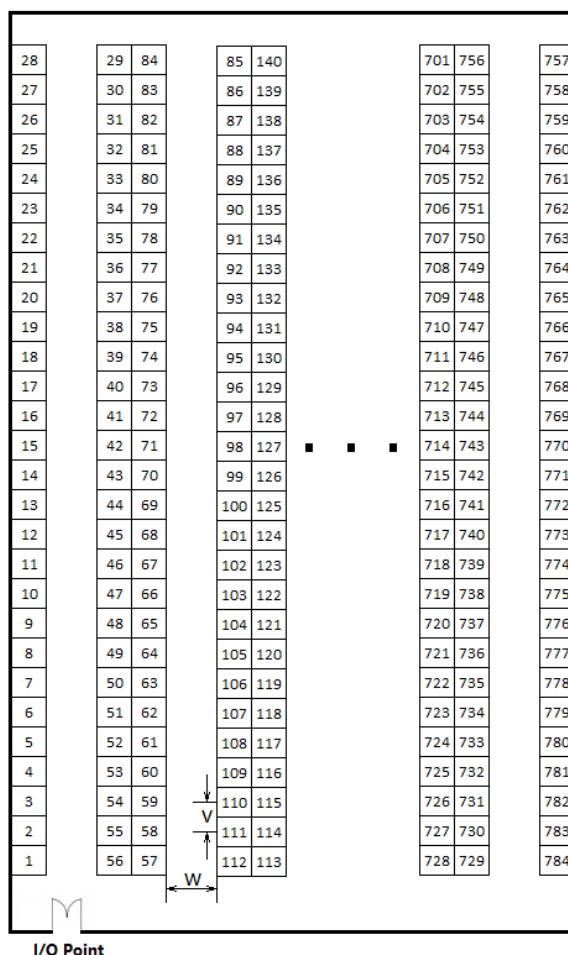


Рисунок 3.8 – План складу з поздовжніми проходами та нумерацією комірок ( $v = 1$  м,  $w = 1,5$  м)

**Позначення та координати.** Проходи індексуємо  $j = 1, \dots, 14$  зліва направо відносно бази (I/O) на ближньому боці складу. Уздовж кожного проходу розглядаємо дискретні позиції з індексом  $l \in \{1, \dots, 28\}$ ; крок між сусідніми позиціями дорівнює  $v = 1$  м, а повна довжина проходу  $h = 28$  м (тобто  $L = h/v$ ).  $y(l)$  – поздовжня відстань від ближнього боку складу до позиції з індексом  $l$  у середині проходу. Вимірюється вздовж осі проходу та не залежить від того на якій стороні стелажа (зліва чи справа) розташована комірка і задає «глибину» позиції. Для крайніх значень  $y(1) = 1$  м,  $y(28) = 28$  м.

$$y(l) = lv, \quad 0 < y(l) \leq h \quad (3.6)$$

Крок між сусідніми проходами уздовж будь-якого торцевого коридору обчислюється як  $s = w + t$ . Горизонтальна координата входу в  $j$ -й прохід уздовж будь-якого торцевого коридору (ближнього або дальнього) однакова й задається

$$x(j) = (j - 1)s \quad (3.7)$$

**Базові формули відстаней.** Наведемо спрощені вирази достатні для подальших розрахунків як під час розміщення так і під час відбору.

**1. Відстань до однієї комірки  $j, l$  туди й назад.** Уздовж торцевих коридорів прийнято один і той самий крок між сусідніми проходами  $s$ . Найкоротший маршрут до комірки  $j, l$  збирач замовлень проходить торцем складу до входу у прохід  $j$ , потім рухається уздовж проходу на відстань  $y(l)$ .

$$D_{cell}(j, l) = 2(x(j) + y(l)) \quad (3.8)$$

**2. Замовлення/партія з кількох позицій.** Нехай задіяна множина проходів  $J^* \subseteq \{1, \dots, 14\}$ , а для кожного задіяного проходу  $j$  найдальша потрібна глибина –  $l_j^{max}$ . Для простої та стабільної оцінки довжину маршруту розкладемо на горизонтальну і вертикальну складові. Горизонтальна частина описує рух торцевим коридором до найдальшого від I/O задіяного проходу і назад, тому

$$D_{hor} = 2(j_{max} - 1)s, \quad j_{max} = \max J^* \quad (3.9)$$

Вертикальна складова дорівнює сумі відстаней, пройдених усередині кожного відвіданого проходу від входу до найбільш віддаленої потрібної позиції та у зворотному напрямку.

$$D_{ver} = \sum_{j \in J^*} 2y(l_j^{max}) \quad (3.10)$$

Загальна оцінка довжини шляху

$$D_{order} = D_{hor} + D_{ver} \quad (3.11)$$

Запропонована конфігурація й формули прив'язуються безпосередньо до параметрів рис. 3.8. Такий підхід забезпечує однозначний і простий механізм підрахунку пройдених відстаней для двох базових операцій, а саме розміщення товарів та комплектування і буде застосований надалі.

### **3.6 Моделювання та порівняння стратегій розміщення товарів на складі**

**Огляд методів організації зберігання.** Проаналізуємо чотири підходи до розміщення SKU на складі, а саме: випадкове розміщення, закріплене розміщення за класами популярності (ABC), **найближча вільна комірка** (*Closest Open Location, COL*) та запропонований кластерний підхід TOASL. Кожна з цих політик має власні механізми та вплив на ефективність складських операцій.

- **Випадкове розміщення** (*Random*) є базовим сценарієм без закріплених адрес за яким кожний прийнятий SKU може бути розміщений у будь-якій доступній комірці за умови дотримання загальних обмежень щодо сумісності, рівня зберігання та габаритів. Така схема моделює склад без систематичного порядку за популярністю чи асоціативністю, тому географія заповнення формується випадковим чином. На практиці виконання має два поширені варіанти. Персонал працює за транспортними дорученнями, які WMS надсилає на радіотермінали/сканери із заздалегідь визначеною вільною адресою з пулу дозволених, або за спрощених регламентів оператор може залишати товар у першій доступній поблизу комірці. Обидві реалізації забезпечують ефективне використання складської площі, проте не враховують ні відстань до I/O, ні клас популярності, ні «сусідство» споріднених SKU, що збільшує довжину маршрутів під час комплектування та розпорошує номенклатуру по складу.

- **ABC-розміщення** – класоорієнтований підхід, за якого номенклатуру поділяють на класи А, В, С залежно від популярності (обороту). Найбільш ходові товари класу А (умовно близько 20 % SKU, що формують  $\approx 80$  % обороту) розміщуються в пріоритетній ближній зоні біля точки I/O, позиції класу В – у проміжних секціях, а класу С – у віддалених зонах складу з нижчою доступністю. Концепція зонування передбачає фіксовані зони або адреси для кожного SKU чи принаймні для відповідного класу, а також підтримує сталість адрес у часі, що спрощує навігацію та поповнення. Це скорочує шлях до товарів із високою частотою відбору, однак не враховує асоціативність попиту, тобто які позиції часто замовляються разом, через що споріднені SKU можуть опинятися в різних зонах. До того ж, потреба резервувати місця під максимальний запас кожного SKU створює значний невикористаний простір і знижує загальний коефіцієнт використання складської площі.
- **COL-розміщення** – стратегія мінімізації шляху за якої кожний товар поміщається в доступну комірку, що має найменшу відстань до точки I/O. Інакше кажучи, під час надходження поставки система або оператор обирає найближчу незайняту локацію і саме туди переміщує товар. Такий підхід добре відтворює реальну поведінку персоналу, орієнтовану на мінімізацію пройденої відстані за кожну операцію. Унаслідок цього склад заповнюється поступово від входу до глибини. Передні ділянки швидко займаються, тоді як віддалені зони довше залишаються вільними. COL-розміщення ефективно скорочує шлях під час розміщення товарів на зберігання, проте не накладає жодної логіки впорядкування номенклатури за типом або популярністю, що може ускладнювати подальше комплектування замовлень через розосередження споріднених SKU [97, 98].
- **TOASL-розміщення** є запропонованою кластерною методикою, що групує асортимент за типом продукції та призначає зонам зберігання відповідні групи товарів. Підхід ґрунтується на кластерному аналізі часових рядів попиту, де після нормування ознак сезонності, тренду та варіативності,

Ward методом формують однорідні групи; їх кількість визначають за силуетним індексом і експертною оцінкою, а самі групи задають просторове зонування складу. У розглянутій моделі 784 SKU згруповано у 16 кластерів за подібністю динаміки попиту та типологією виробів. Серед характерних груп – пакети й сумки, кухонні набори та банки, святковий декор, текстиль і столовий посуд, освітлення та свічники, плетені вироби та кошики, дверні килимки й таблички. Для кожного кластера визначається логічна зона зберігання, тобто множина адрес комірок зарезервованих під товари відповідного типу. Оскільки класифікація SKU може змінюватися, конфігурація цих зон є динамічною і переналаштовується на рівні адресації WMS без втручання у фізичне планування складу. При цьому всередині кожної зони застосовується локальний принцип ABC, згідно з яким найпопулярніші SKU свого кластера займають позиції ближче до входу, а менш популярні розташовуються у віддаленій частині відповідної секції зберігання. Таким чином, TOASL поєднує елементи розміщення за класами ABC та тематичного зонування, формуючи кластери з товарів, які часто замовляються разом. Компактне формування запасів скорочує шлях комплектування завдяки зосередженню споріднених позицій поруч. При цьому зберігається ефективність операцій розміщення [83].

**Псевдокод алгоритму TOASL.** Нижче подано узагальнену процедуру адресації зберігання за методом TOASL, що базується на кластеризації попиту та тематичному зонуванні. Вхідними даними є результати кластеризації номенклатури на 16 груп та належність кожного SKU до відповідного кластера. Також використовуються індекси попиту для впорядкування позицій і перелік комірок із відстанями до точки I/O (мангеттенська метрика). Для кожного кластера визначено логічну зону зберігання, тобто множину адрес, зарезервованих під типові товари цієї групи. Межі таких зон є динамічними та можуть переналаштовуватись без зміни фізичного планування. Усередині зони застосовується локальний принцип ABC за яким позиції з вищою популярністю отримують ближчі адреси.



Склад досліджуваного асортименту дає підстави виділяти характерні кластери на кшталт пакети й сумки, кухонні набори та банки, святковий декор, текстиль і столовий посуд, дверні килимки та таблички тощо. Саме такі однорідні групи формують осередки попиту, для яких алгоритм резервує компактні блоки адрес, починаючи від входу, з невеликим резервом місткості під поповнення. Кожна поставка обробляється послідовно. Система визначає кластер для відповідного SKU, обирає найближчу вільну адресу у виділеній зоні та фіксує призначення. Якщо зона переповнена, її розширюють уздовж власної межі. Коли суміжних вільних місць немає, застосовують глобально найближчу адресу. Як альтернатива виділяють частину периферійних комірок із зони кластера нижчого пріоритету.

### Алгоритм TOASL динамічної адресації складу

Вхід:

SKU – множина усіх позицій  
 cluster\_of[sku] – кластер кожного SKU ( $k = 1..K$ )  
 popularity[sku] – показник популярності (для локального ABC)  
 Cells – всі комірки складу, впорядковані за dist\_to\_IO  
 buffer\_ratio  $\in [0,1]$   
 overflow\_rule  $\in \{\text{adjacent, closest, borrow}\}$

Вихід:

Place[sku]  $\rightarrow$  cell

1) Агрегування за кластерами

для  $k=1..K$ :

SKUs[k] := {sku | cluster\_of[sku]=k}  
 demand[k] :=  $\sum$  popularity[sku] по SKUs[k]  
 size[k] := |SKUs[k]|

2) Порядок обробки

Order := кластери k, відсортовані за спаданням demand[k]

3) Формування динамічних зон

Free := Cells

для  $k \in$  Order:

cap := ceil(size[k]\*(1+buffer\_ratio)) # cap – цільова місткість зони, тобто кількість комірок, необхідна для розміщення SKU кластера з урахуванням резерву  
 Zone[k] := TakeNearestContiguous(Free, cap) # суцільний блок(и) близько до I/O

Free := Free \ Zone[k]

4) Розміщення в межах зони (локальний ABC)

для  $k \in \text{Order}$ :

$\text{Addr} := \text{Zone}[k]$  відсортовані за зростанням  $\text{dist\_to\_IO}$

$\text{Items} := \text{SKUs}[k]$  відсортовані за спаданням  $\text{popularity}$

$i := 1$

для  $\text{sku} \in \text{Items}$ :

якщо  $i \leq |\text{Addr}|$ :

$\text{Place}[\text{sku}] := \text{Addr}[i]; i := i+1$

інакше:

$\text{Place}[\text{sku}] := \text{HandleOverflow}(\text{Free}, \text{Zone}[k], \text{overflow\_rule})$

позначити призначену комірку як зайняту та вилучити з  $\text{Free}$

## 5) Повернути Place

# Допоміжні процедури

$\text{TakeNearestContiguous}(\text{Free}, \text{cap})$ :

# Ітеративно, крок за кроком, обирає найближчі до I/O суміжні комірки, доки не буде набрано cap адрес.

# Перевага надається послідовним адресам в одному проході.

...

$\text{HandleOverflow}(\text{Free}, \text{Zone}, \text{rule})$ :

якщо  $\text{rule}=\text{adjacent}$ : повернути найближчу до межі  $\text{Zone}$  вільну комірку (розширити зону)

якщо  $\text{rule}=\text{closest}$ : повернути глобально найближчу вільну комірку

якщо  $\text{rule}=\text{borrow}$ : позичити найдальшу комірку в зони кластера з найнижчим пріоритетом

### 3.6.1 Методика моделювання процесу розміщення товарів

Для порівняння описаних політик розміщення виконано серію експериментів у симуляційному середовищі, яке відтворює геометрію складу, правила обчислення відстаней та режим роботи комплектувальника. План експериментального моделювання визначає ключові умови, припущення й обмеження на яких базується подальший аналіз ефективності кожної політики.

**1. Генерація та структура поставок.** Кожен експеримент починається з порожнього складу, після чого моделюється послідовність із 10–15 поставок. Номенклатура кожної партії визначається як випадковою вибіркою з переліку 784 SKU, так і об'єднанням споріднених позицій у межах одного надходження, що відповідає товарам від одного постачальника. У моделюванні поєднуються обидва підходи, щоб забезпечити різноманітність складу між окремими експериментальними

запусками. Для кожного SKU в поставках кількість прийнятих товарів становить 100 одиниць і вони розміщуються в одній комірці.

2. **Маршрут переміщення.** Кожен маршрутний цикл стартує в точці I/O і завершується поверненням до неї. Пересування під час розміщення моделюється дискретно на сітці плану складу (рис. 3.8), рух дозволено лише вздовж проходів і торцевих коридорів. Відстані обчислюються за мангеттенською метрикою як сума ортогональних відрізків. Для кожного переходу від поточної позиції до заданої комірки вибирається найкоротший можливий шлях, тобто забезпечується локальна оптимальність без глобальної перебудови маршруту. Порядок відвідування комірок не оптимізується і відповідає послідовності надходження товарів. Спеціалізовані алгоритми маршрутизації не застосовуються. Виконавець моделюється як агент, що діє цілеспрямовано та уникає зайвих зворотних рухів, забезпечуючи наближену мінімізацію шляху в межах кожного маршрутного циклу. Така постановка дозволяє оцінити мінімально необхідну сумарну довжину переміщень для різних політик, не прив'язуючи результати до конкретних правил глобальної маршрутизації.
3. **Організація процесу розміщення.** Усі переміщення виконує один працівник, який опрацьовує поставки послідовно. Паралельна робота кількох комплектувальників і можливі затримки через вузькі проходи не враховуються, взаємні блокування у проходах не виникають.
4. **Розміщення товарів.** Приймається режим унітарного зберігання при якому кожна комірка вміщує лише один SKU. Вважається, що комплектувальник виконує розміщення 5 різних товарних позицій за один обхід (цикл). Під час одного маршрутного циклу працівник послідовно поміщає всі 5 SKU на місця зберігання, після чого повертається до I/O для можливого отримання наступної партії товарів. У випадку політики ABC для кожного SKU попередньо закріплено фіксовану зону/комірку відповідно до його класу, для методу TOASL – зону відповідного кластера. Для Random і COL пріоритетні зони не встановлюються, адреси вибираються серед наявних

вільних комірок згідно з правилами відповідної схеми. Усі призначення формує WMS і надсилає як транспортні доручення на сканер оператора.

**5. Повторення експериментів.** Для кожної політики виконано 20 незалежних симуляційних запусків, щоб згладити випадкові коливання. Повну довжину шляху, який проходить комплектувальник під час розміщення всієї номенклатури, обчислено у межах кожного експериментального запуску. Результати усереднено для отримання узагальненої оцінки ефективності. У підсумку отримано середнє значення сумарного шляху у метрах необхідного для розміщення всієї номенклатури за обраною політикою зберігання.

У моделі не враховуються логістичні характеристики товарів, як-от габаритні розміри, маса, вимоги до умов зберігання та упаковки. Передбачається однорідність SKU за розміром, тож їх можна розміщувати в будь-якій комірці. Таке спрощення є прийнятним, оскільки зазначені параметри зазвичай контролюються WMS і не впливають безпосередньо на довжину переміщень під час розміщення. Завдяки цьому аналіз зосереджено на порівнянні сумарних довжин маршрутів для різних політик зберігання.

### **3.6.2 Результати моделювання та порівняльний аналіз політик зберігання**

У таблиці 3.2 наведено узагальнені результати порівняння чотирьох політик. Подано середню сумарну відстань, яку проходить комплектувальник під час розміщення товарів усіх поставок, а також відносне відхилення цього показника від базового рівня, причому TOASL прийнято за 100 %. Менше значення дистанції відповідає ефективнішому (коротшому) процесу розміщення товарів на зберігання. Для інтерпретації використано відсоткове відхилення, яке обчислюється за формулою

$$\frac{D_{policy} - D_{TOASL}}{D_{TOASL}} \cdot 100\% \quad (3.12)$$

де  $D_{policy}$  – сумарний шлях для розглядуваної політики зберігання, усереднений за виконаними експериментами,  $D_{TOASL}$  – відповідний середній сумарний шлях для політики TOASL. Від’ємне відхилення означає кращий (коротший) результат порівняно з TOASL, додатне – гірший.

Таблиця 3.2 – Сумарний шлях комплектувальника при розміщенні товарів та порівняння стратегій

Політика розміщення	Сумарний шлях, м	Відхилення від TOASL, %
Random	28915	9,2
ABC	30112	13,7
COL	25084	-5,3
TOASL	26476	

**Аналіз результатів.** Наведені цифри демонструють чіткі відмінності між досліджуваними підходами. Найменшу сумарну відстань забезпечує стратегія найближчого вільного місця  $\approx 25,1$  км на весь обсяг розміщення, що на 5,3 % менше від базового рівня. Це очікувано, адже стратегія локально мінімізує пройдений шлях, тому кожен товар розміщується в найближчій доступній комірці. Відповідно кожен маршрутний цикл комплектувальника має мінімальну довжину, а сукупні витрати часу та пройденої відстані на розміщення знижуються. Натомість метод COL не забезпечує впорядкування номенклатури за типами. Товари розосереджуються випадковим чином і здебільшого накопичуються поблизу входу. Хоча це вигідно для процесу розміщення, відсутність організації може негативно вплинути на подальший відбір замовлень, оскільки пов’язані товари можуть опинитися далеко один від одного.

Сумарний шлях за TOASL становить  $\approx 26,5$  км і приймається за базовий рівень 100 %. TOASL поступається COL за ефективністю розміщення, оскільки середня дистанція більша приблизно на 5 %, що є наслідком дотримання обмежень зонування. За цієї політики комплектувальник іноді змушений пройти трохи далі до відповідної зони кластера замість того, щоб

залишити товар у найближчій доступній комірці. Однак різниця з COL відносно невелика, тож кластерний підхід і надалі забезпечує короткий маршрут до місць зберігання товарів. Поза тим, додаткові витрати шляху можуть компенсуватися на етапі комплектування замовлень. Оскільки товари одного типу згруповано, зростає ймовірність, що під час збирання багатопозиційної заявки потрібні SKU будуть розташовані в межах однієї зони.

Випадкове розміщення дало сумарну дистанцію близько 28,9 км, що на 9,2 % більше від TOASL і є третім за ефективністю результатом. Попри уявну простоту, стратегія Random не дає переваг для розміщення, оскільки адреси призначаються без критерію мінімальної відстані. Транспортні доручення формує WMS і надсилає їх на радіотермінал комплектувальника, тому частина позицій може опинитися на віддалених локаціях. Водночас певна гнучкість випадкового розміщення інколи дозволяє розташовувати малопопулярні позиції ближче до I/O, ніж це було б за жорсткої ABC-схеми. Це пояснює чому ABC-розміщення продемонструвало найгірший результат, близько 30,1 км, що на 13,7 % більше за базовий рівень. За ABC політика є суворо регламентованою, для кожного SKU визначається фіксована адреса, а для малопопулярних позицій такі місця зазвичай розташовуються у віддалених секціях складу. Тому навіть рідкі поставки класу C змушують комплектувальника щоразу долати максимальну відстань. Політика Random позбавлена цього недоліку, оскільки систематичне віднесення малопопулярних товарів на периферію не відбувається, тож середній шлях дещо менший ніж за ABC. Водночас підхід ABC має перевагу, адже розміщення товарів класу A ближче до входу потенційно знижує трудомісткість комплектування популярних замовлень. Втім, як уже зазначалося, ABC не враховує структуру спільного попиту. Товари, які часто купують разом, можуть опинитися в різних зонах, скажімо один у зоні A, інший у зоні C, що зрештою подовжує маршрут комплектування. Кластерний метод TOASL спеціально покликаний усунути цей недолік, оскільки

групування взаємопов'язаних SKU формує більш «компактні» набори товарів і забезпечує виграш під час здійснення відбору багатоасортиментних замовлень.

Результати експериментального моделювання підтверджують важливість вибору раціональної політики розміщення. Найшвидший розподіл по складським місцям забезпечує політика найближчої вільної комірки, однак такий підхід не оптимізує подальше комплектування. Класифікація за популярністю згідно зі схемою ABC сама по собі недостатня і навіть може підвищувати трудомісткість розміщення. Запропонована стратегія TOASL має збалансований характер і лише незначно, приблизно на 5 %, поступається найкращому випадку під час розміщення товарів на зберігання, зате створює передумови для більш ефективного відбору товарів. Таким чином, TOASL можна розглядати як компромісне рішення, що поєднує прийнятні оперативні витрати часу на розміщення з покращенням глобальної ефективності виконання замовлень на складі.

### 3.7 Стратегії комплектування замовлень на складі

**Систематизація підходів до відбору замовлень на складі.** Ефективність збирання замовлень значною мірою залежить від застосованої стратегії комплектування. Обрана схема визначає, як саме розподіляються замовлення між маршрутами відбору і як організовано рух комплектувальника на складі, доповнюючи політику маршрутизації. Існують чотири основні підходи до відбору товарів, а саме: **комплектування одиничних замовлень** (*Discrete picking*), **комплектування партіями** (*Batch picking*), **кластерне комплектування** (*Cluster picking* або *sort-while-pick*) та **зональне комплектування** (*Zone picking*) [8]. Кожна з цих стратегій має свої особливості, переваги та недоліки, що впливають на загальну продуктивність складу. Розглянемо сутність зазначених підходів та наведемо їх порівняння.

- **Комплектування одиничних замовлень** є базовою й найпоширенішою стратегією за якої кожне замовлення збирається окремим комплектувальником відповідно до індивідуального списку [8]. Серед її переваг простота, надійність і мінімальний ризик плутанини, а також можливість розпочинати нове термінове замовлення одразу після надходження. Головний недолік полягає в низькій продуктивності, оскільки кожен маршрут обслуговує лише один набір товарних позицій. За великої номенклатури та розосередженого зберігання це спричиняє зайві переміщення й значні витрати часу, що подекуди становлять більшу частину операційних витрат [99]. Отже, підхід виграє в точності, але поступається ефективністю при великих потоках дрібних замовлень.
- **Комплектування партіями** – стратегія за якої кілька окремих замовлень об'єднуються в одну партію і відбираються одночасно в ході єдиного маршруту [100]. Комплектувальник отримує зведений список, що охоплює товари одразу з кількох замовлень, і здійснює єдиний обхід складу для збору всіх позицій. Ключова ідея полягає у скороченні дублювання маршрутів. Кожна комірка відвідується лише раз для відбору всіх потрібних товарів, навіть якщо вони належать до різних замовлень. Це підвищує продуктивність і зменшує загальну довжину маршруту, особливо коли різні замовлення містять спільні або розташовані близько одна до одної позиції. Втім, істотним недоліком batch-стратегії є потреба подальшого розподілу товарів за окремими замовленнями. Під час відбору товари різних клієнтів збираються разом у спільний контейнер або на візок, тому після повернення необхідний етап сортування, щоб розкласти їх за відповідними замовленнями. Таким чином, за такого підходу частина роботи фактично переноситься на етап після збирання, тож скорочення шляху оператора супроводжується додатковими витратами часу на сортування й консолідацію замовлень перед відправкою. Поза тим, великі партії затримують виконання окремих заявок до моменту формування та



ускладнюють контроль якості, підвищуючи ризики неправильного сортування товарів.

- **Кластерне комплектування** – це вдосконалення партійного підходу, яке дозволяє уникнути окремого етапу сортування. Кластерна стратегія передбачає одночасний збір кількох клієнтських заявок, як і у batch-режимі, але з тією різницею, що товари безпосередньо під час збирання для кожного замовлення розкладаються по окремих контейнерах на спеціальному візку. Тобто реалізується принцип за якого відбувається сортування під час відбору. Комплектувальник вирушає на маршрут з візком, обладнаним декількома відсіками для розміщення товарів різних заявок. Діставшись локації відбору, він одразу розкладає відібрані товарні одиниці у відповідні відсіки, закріплені за кожною заявкою. Таким чином, досягнувши точки завершення маршруту, працівник має на візку вже розсортовані замовлення готові до пакування без подальшої обробки на сортувальному столі. Кластерне комплектування по суті поєднує переваги партійного методу, зокрема скорочення сукупного шляху завдяки спільному відбору замовлень із мінімізацією операцій після повернення. Кількість заявок у кластері обмежується місткістю візка, тобто числом окремих контейнерів відведених під кожну з них [101]. Зазвичай використовують візки на 4–12 замовлень, щоб, з одного боку, суттєво підвищити продуктивність маршруту відбору, а з іншого не ускладнювати процес надмірною кількістю паралельно оброблюваних замовлень. На практиці кластерний підхід найкраще працює за великої кількості дрібних клієнтських заявок характерних для інтернет-торгівлі. Замість окремого відбору кожної з них, яка часто містить лише 1–2 позиції, комплектувальник за один прохід збирає одразу кілька невеликих комплектів. Дослідження підтверджують, що впровадження такого відбору здатне суттєво підвищити продуктивність складу у порівнянні з дискретною стратегією. Водночас кластерне комплектування потребує належної організації. Потрібні спеціальні візки або багатосекційні контейнери. Важливий чіткий інструктаж персоналу, щоб уникати помилкового

покладання товарів у невідповідні відсіки. Також необхідний ефективний алгоритм формування самих кластерів, тобто груп заявок. Бажано добирати їх з урахуванням близькості та спільних маршрутів, тоді скорочення дистанції буде максимальним. Попри ці вимоги, переваги кластерної стратегії є суттєвими, і саме вона нині широко використовується провідними e-commerce операторами для прискорення обробки онлайн-замовлень.

- **Зональне комплектування** – стратегія просторового поділу складу на зони відбору. У межах цієї стратегії кожен комплектувальник закріплений за своєю ділянкою й відбирає лише товари, що там зберігаються. Водночас окреме замовлення нерідко охоплює кілька зон, тому потрібна узгоджена координація між ними. Існують два різновиди зонального комплектування, а саме **послідовний** (*pick-and-pass*), за якого контейнер передають від зони до зони, і **паралельний**, у межах якого відбір виконують паралельно в кількох зонах, а зібрані частини згодом консолідують у єдиний набір [102]. До переваг такого підходу належать скорочення шляху для кожного працівника, зростання швидкості й точності завдяки кращому орієнтуванню у власній зоні, а також підвищення пропускної здатності, оскільки операції виконують паралельно. Серед недоліків варто відзначити складнішу координацію й консолідацію. У послідовному варіанті можливі простоя та дисбаланс навантаження, тоді як у паралельному зростають вимоги до відстеження і часто виникає потреба у сортувальних системах чи конвеєрах. Метод доцільний для великих дистрибуційних центрів і складів із широким асортиментом, оскільки зонування локалізує переміщення, зменшує зустрічні потоки та дає змогу спеціалізувати персонал на певних групах товарів.

**Порівняння стратегій.** Для наочності узагальнимо в таблиці ключові характеристики описаних підходів, порівнявши способи відбору, потребу в сортуванні та основні переваги й недоліки кожної стратегії [8, 100].

Таблиця 3.3 – Порівняльна характеристика стратегій комплектування замовлень

Стратегії	Характеристика відбору товарів	Переваги	Недоліки
Discrete [8]	Кожне замовлення збирають окремо одним маршрутом відбору	Простота, мінімум помилок, можливість одразу збирати термінові заклази	Низька продуктивність, багато зайвих переміщень
Batch [100]	Кілька замовлень об'єднують у партію, відбирають разом; сортування виконують після повернення	Менше проходів по складу, вища продуктивність	Потрібне сортування після відбору, можливі затримки до формування партії, ризик плутанини товарів
Cluster [101]	Одноточасний відбір кількох замовлень з розкладанням у відповідні комірки візка	Скорочення шляху без окремого сортування, висока швидкість для дрібних e-commerce замовлень, ефективніше використання робочого часу персоналу	Обмеження місткістю візка, можливі труднощі з добором замовлень до складу кластера
Zone [102]	Склад поділено на зони, відбір виконують у своїх зонах, після частини замовлення консолідуєть	Коротші переходи, спеціалізація працівників, паралельність підвищує пропускну здатність	Складна координація й консолідація, можливі простоти та дисбаланс, інколи потрібні конвеєри/сортери

Як показано у табл. 3.3, дискретна стратегія є надійною та простою у реалізації, проте створює найбільше навантаження на комплектувальників через повторні обходи складу для кожного замовлення. Партійний та кластерний методи дозволяють істотно зменшити сукупний шлях руху, об'єднуючи кілька замовлень у межах одного маршруту відбору [100]. При

цьому кластерне комплектування має вирішальну перевагу над звичайним batch-підходом через відсутність окремого етапу сортування після збирання, що скорочує загальний цикл обробки замовлень [101]. У сучасних умовах, коли йдеться про виконання щодоби сотень і тисяч онлайн-замовлень, кластерна стратегія дозволяє підвищити продуктивність складу на 15–40 % завдяки скороченню часу переміщень і швидкій готовності зібраних позицій до відправлення порівняно з дискретним методом. Залежно від структури замовлень та планування маршрутів це зростання може бути ще більшим. Зональне комплектування, у свою чергу, адресує проблему великого асортименту і площі складу, забезпечуючи паралельне виконання операцій відбору. Його доцільно застосовувати у великих розподільчих центрах, хоча впровадження потребує ретельного планування зон та додаткових технічних рішень для подальшої консолідації потоків.

**Вибір стратегії для дослідження.** З огляду на вищенаведений аналіз у подальших експериментах буде застосовано кластерне комплектування як єдину досліджувану стратегію. Вона найбільше відповідає умовам електронної комерції з високою інтенсивністю надходження невеликих заявок та жорсткими часовими вікнами виконання. Кластерний підхід зменшує трудомісткість відбору одиничних позицій і забезпечує короткий цикл виконання від прийняття замовлення до відвантаження, завдяки об'єднанню маршрутів і відсутності окремого етапу сортування [101]. У межах даного дослідження на основі базової політики кластерного комплектування сформовано програму експериментів та критерії оцінювання ефективності. Обрана стратегія інтегрується з методикою TOASL. Її буде використано для оцінювання впливу правил групування заявок і схем маршрутизації на довжини маршрутів відбору.

### 3.8 Теоретичні засади Box Plot, однофакторної ANOVA та критерію Тьюкі HSD

Науково обґрунтуємо опис статистичного інструментарію, застосованого для порівняння стратегій розміщення товарів Random, ABC, COL і TOASL за п'ятьма маршрутами кластерного збирання S-shape, mid-point, Largest Gap, Return та Composite. Метою є коректна оцінка впливу обраної політики розміщення на довжину шляху комплектувальника та перевірка відмінностей між середніми значеннями відстаней.

Основною вихідною величиною є довжина маршруту в метрах, необхідна для виконання одного циклу комплектування під час збирання партії замовлень. Для кожної комбінації маршруту та політики реєстрували незалежні спостереження. У дослідженні виконано багаторазові незалежні вимірювання на кількох змодельованих поставках за фіксованої геометрії складу з поздовжніми проходами та фронтальною точкою I/O. Дистанцію обчислювали як суму горизонтальної та вертикальної складових відповідно до правил кожного маршруту. Геометрія складу, система позначень і положення точки I/O, використані в розрахунках відстаней, подані на рис. 3.8, де також наведено схему розміщення рядів і нумерацію місць зберігання.

Для первинної діагностики варіабельності та виявлення потенційних аномалій застосовано **діаграму розмаху** (*box plot*), яка відображає медіану, квартилі  $Q_1$ ,  $Q_3$ , міжквартильний розмах  $IQR$  і «вуса». Таке подання забезпечує наочне зіставлення розподілів відстаней між політиками в межах кожного маршруту та дозволяє ідентифікувати віддалені спостереження [103].

$$Q_2 = \text{median}(y), \quad Q_1 = 25\text{й перцентиль}, \quad Q_3 = 75\text{й перцентиль}. \quad (3.13)$$

Межі «вусів», а також нижню й верхню межі визначасмо за класичними правилами:

$$Q_1 - 1,5IQR, \quad Q_3 + 1,5IQR. \quad (3.14)$$

Формальну перевірку відмінностей між варіантами розміщення виконуємо за допомогою однофакторного дисперсійного аналізу ANOVA, де єдиним фактором виступає «політика розміщення». Нехай  $y_{ij}$  – відстань  $j$ -го спостереження у  $i$ -тій групі;  $n_i$  – обсяг  $i$ -тої групи;  $k$  – кількість груп;  $N = \sum_{i=1}^k n_i$ . Позначимо групові та загальне середні як  $\bar{y}_i$  і  $\bar{y}$ . За цих позначень міжгрупова та внутрішньогрупова суми квадратів, відповідні середні квадрати, а також статистика Фішера мають вигляд:

$$\bar{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^k \sum_{j=1}^{n_i} y_{ij} \quad (3.15)$$

$$SSB = \sum_{i=1}^k n_i (\bar{y}_i - \bar{y})^2, \quad (3.16)$$

де  $SSB$  – міжгрупова сума квадратів, що відображає варіацію між груповими середніми відносно загального середнього.

$$SSW = \sum_{i=1}^k \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2, \quad (3.17)$$

де  $SSW$  – внутрішньогрупова сума квадратів, що характеризує варіацію всередині груп навколо їхніх власних середніх.

$$SST = SSB + SSW, \quad (3.18)$$

де  $SST$  – повна сума квадратів (загальна варіація даних).

$$df_B = k - 1, \quad df_W = N - k, \quad (3.19)$$

де  $df_B$  – ступені вільності між групами, що визначаються кількістю груп  $k$ , а  $df_W$  – ступені вільності всередині груп, визначені загальною кількістю спостережень  $N$  та числом груп.

$$MSB = \frac{SSB}{df_B}, \quad MSW = \frac{SSW}{df_W}. \quad (3.20)$$

$$F = \frac{MSB}{MSW} \sim F_{(df_B, df_W)}. \quad (3.21)$$

Нульова гіпотеза  $H_0$  у нашому контексті означає відсутність ефекту політики розміщення, тобто всі групові середні однакові ( $\bar{y}_1 = \dots = \bar{y}_k$ ). Рівень значущості  $\alpha$  – це прийнята наперед ймовірність помилки першого роду (відхилення істинної  $H_0$ ), у дослідженні  $\alpha = 0,05$ . Рішення ухвалюємо за р-значенням і якщо  $p < \alpha$ , нульову гіпотезу відхиляємо й визнаємо, що принаймні дві середні різняться. Після відхилення  $H_0$  застосовуємо множинні попарні порівняння за **критерієм Тьюкі** (*Tukey's Honestly Significant Difference, HSD*), який контролює **групову ймовірність помилки першого роду** (*family-wise error rate*) та надає скориговані р-значення і 95 % довірчі інтервали для різниць середніх.

Після оцінювання дисперсійних компонентів доцільно подати міру ефекту, що показує, яку частку загальної варіації пояснює фактор «політика розміщення». Використовуємо частку поясненої дисперсії  $\eta^2$  – частку загальної варіації, яку можна приписати фактору «політика розміщення», а також її скориговану, менш зміщену щодо розміру вибірки оцінку  $\omega^2$ , що зменшує завищення ефекту на малих вибірках [104].

$$\eta^2 = \frac{SSB}{SST}. \quad (3.22)$$

$$\omega^2 = \frac{SSB - df_B \cdot MSW}{SST + MSW} \quad (3.23)$$

Якщо нульову гіпотезу рівності середніх відхилено, переходимо до множинних попарних порівнянь за процедурою Tukey HSD. Для незбалансованого плану стандартну помилку різниці двох середніх визначаємо як

$$SE_{ij} = \sqrt{MSW \left( \frac{1}{n_i} + \frac{1}{n_j} \right)}. \quad (3.24)$$

Критичне значення визначаємо за розподілом нормованого розмаху (Studentized range)  $q$  для  $k$  груп і  $df_W$  ступенів вільності [105], тож поріг HSD дорівнює:

$$q_{\alpha;k,df_W} \Rightarrow HSD_{ij} = q_{\alpha;k,df_W} \cdot SE_{ij} \quad (3.25)$$

Рішення для пари (i,j) формулюємо через правило відсікання:

$$|\bar{y}_i - \bar{y}_j| > HSD_{ij} \Leftrightarrow q_{ij} = \frac{|\bar{y}_i - \bar{y}_j|}{SE_{ij}} > q_{\alpha;k,df_W}. \quad (3.26)$$

Довірчий інтервал для різниці середніх на рівні довіри  $1 - \alpha$  має вигляд:

$$[(\bar{y}_i - \bar{y}_j) - q_{1-\alpha;k,df_W} SE_{ij}, (\bar{y}_i - \bar{y}_j) + q_{1-\alpha;k,df_W} SE_{ij}]. \quad (3.27)$$

Цього підходу послідовно дотримуємося для всіх діаграм розмаху у кожному маршруті та для всіх політик розміщення. Така сукупна послідовність аналізу починається з візуалізації через box plot, далі виконується однофакторний дисперсійний аналіз ANOVA з оцінками  $\eta^2$  та  $\omega^2$ , а потім – множинні порівняння Tukey HSD. Такий підхід забезпечує водночас наочність інтерпретації та строгість статистичного висновку щодо відмінностей між політиками Random, ABC, COL, TOASL у всіх п'яти маршрутах S-shape, mid-point, Largest Gap, Return, Composite.

### 3.8.1 Методологія експериментальних досліджень

**Конфігурація складу та геометрія.** Моделюється прямокутний склад із паралельними проходами та двома торцевими коридорами, а точка I/O розташована на ближньому (фронтальному) боці. План і позначення відповідають схемі на рис. 3.8, яка фіксує поздовжній крок сітки  $v = 1$  м, довжину проходу  $h = 28$  м, відстань між входами сусідніх проходів  $w = 1,5$  м і поперечний крок  $m = 2$  м. На їх основі визначається система координат (адресації) комірок, яка використовується для розрахунку горизонтальних та



вертикальних складових шляху. Переміщення здійснюються лише вздовж проходів і торцевих коридорів, причому довжину шляху обчислюємо за мангеттенською метрикою як суму ортогональних відрізків між I/O та позиціями відбору товарів.

**Дані, кластери та політика TOASL.** Експерименти базуються на наборі даних E-commerce dataset, структуру якого детально описано та проаналізовано вище. У дослідженні використано 784 SKU, що забезпечують близько 80 % виторгу. На підставі часових рядів продажів виконано попередню обробку даних, яка включала очищення транзакцій від аномалій і службових записів, побудову нормованих рядів попиту для кожного SKU з масштабуванням до  $[0;1]$  та заповненням відсутніх значень нулями. Відтак застосовано агломеративну кластеризацію методом Ward. У результаті номенклатуру згруповано у 16 кластерів. Надалі ці групи використовуються в гібридній політиці TOASL для зонування та локального ABC-впорядкування всередині зон.

**Генерація поставок і замовлень.** Експеримент запускається з порожнього складу. Для кожної політики розміщення товарів моделюємо послідовність із 10 поставок. Для кожного SKU надходить по 100 одиниць в одну комірку. Оцінюються факт відвідування комірки та сумарна довжина пройденого шляху. Вертикальні операції підйому не моделюються, оскільки розглядається одноярусне зберігання. Щоб отримати реалістичні партії замовлень, формуємо замовлення на 5, 10, 15, 20 та 25 позицій. Базою слугує тематична структура номенклатури з 16 груп (кластерів). Для кожної партії довільно обирається фокусна група – кластер, що домінує у складі відповідного замовлення та об'єднує товари зі спорідненими профілями попиту чи призначенням. З цієї групи відбираємо 60–80 % позицій. Решту 20–40 % додаємо як контрольний «шум», добираючи товари випадковим чином з інших кластерів із рівною ймовірністю. Таке змішування відтворює реальні міжкатегорійні замовлення та запобігає надмірно жорстким зв'язкам між

кластером і маршрутом. Воно також дає змогу оцінити стійкість політик розміщення й методів побудови маршрутів.

У дослідженні реалізовано чотири політики розміщення: Random (випадкове призначення), ABC (зонування за інтенсивністю попиту з наближенням класу А до I/O), COL (найближча вільна комірка) та TOASL – запропонована гібридна політика кластерного зонування з локальним ABC. У поєднанні з кожною з перелічених політик розміщення маршрути комплектування будуються за п'ятьма правилами S-shape, mid-point, Largest Gap, Return і Composite. Їх формальні визначення та алгоритмічні кроки наведено вище, і вони відповідають загальноприйнятим підходам при партійному комплектуванні замовлень. Планування експерименту та набір факторів визначено у форматі повного перехресного експерименту. Дизайн експериментів включає 10 поставок, 4 політики розміщення, 5 типів маршрутів комплектування та 5 варіантів розміру партії замовлень. Для кожної комбінації виконано по два незалежні повторення, що загалом дало 2000 спостережень. Основна результативна змінна – довжина маршруту, виміряна в метрах, на один цикл збирання партії замовлень. Запропонований опис узгоджує геометрію складу, джерело даних і процедуру кластеризації товарів, а також спосіб формування партій замовлень, набір політик і маршрутів та правила обчислення відстані. Така структуризація усуває дублювання та акцентує увагу на відтворюваності експерименту.

### **3.9 Експериментальні результати та їх статистична оцінка**

Наведено результати моделювання, які засвідчують ефективність різних комбінацій стратегій комплектування та політик розміщення товарів. Аналіз виконується для кожного сценарію окремо із поданням даних на коробкових графіках («ящиках з вусами») та перевіркою статистичної значущості за однофакторним ANOVA і пост-хок тестом Тьюкі. В усіх експериментах кожену позицію замовлення відбирають у кількості 1 шт., оскільки цікавить

насамперед факт відвідування комірки (осередку зберігання), а не обсяг відбору. Для відповідної конфігурації виконується серія повторів й подається звіт із середніми значеннями отриманих показників.

### **3.9.1. S-shape: стратегія кластерного комплектування та порівняння політик розміщення**

Розглянемо вплив обраного методу розподілу товарів на дистанцію переміщення персоналу за маршрутом S-shape у режимі кластерного відбору (рис. 3.1, а). За цього сценарію відвідується кожен прохід із позиціями замовлення і відбувається повернення до точки I/O від останнього відвіданого, а проходи без відборів оминаються. Для кожного варіанта організації зберігання сформовано розподіли довжин маршрутів, далі виконано однофакторний дисперсійний аналіз (ANOVA) та, за наявності значущості, пост-хок порівняння за критерієм Тьюкі з поправкою на множинні порівняння.

На рис. 3.9 наведено коробковий графік показників пройденого шляху для політик Random, ABC, COL та TOASL. Візуально спостерігається найменша медіана і відносно вузький міжквартильний діапазон («коробка») у TOASL, що узгоджується з гіпотезою про ефект часово-орієнтованого розміщення. Для ABC медіана також нижча, ніж у Random і COL. Водночас Random та COL демонструють вищі медіани і ширший інтерквартильний розмах, що вказує на більшу варіативність основної маси значень.

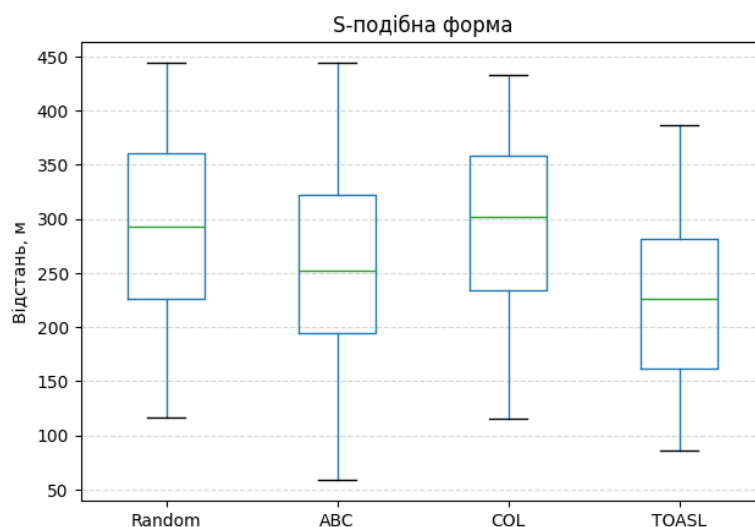


Рисунок 3.9 – Box plot довжин маршрутів для політик Random, ABC, COL і TOASL за маршрутом S-shape

Таблиця 3.4 – Середні дистанції (м) за різної кількості позицій для першого сценарію. У дужках позначено відсоткову зміну відносно політики Random

К-сть позицій	Random	ABC	COL	TOASL
5	160,2	137,4 (-14 %)	164,6 (+3 %)	125,0 (-22 %)
10	246,0	223,4 (-9 %)	246,6 (0 %)	200,5 (-18 %)
15	312,2	278,8 (-11 %)	316,0 (+1 %)	249,0 (-20 %)
20	352,0	290 (-18 %)	345,0 (-2 %)	262,0 (-26 %)
25	391,6	343,8 (-12 %)	372,2 (-5 %)	308,0 (-21 %)

За всією вибіркою середня довжина маршруту становить 292,4 м для Random, 254,7 м для ABC, 288,9 м для COL і 228,9 м для TOASL. Результати однофакторного ANOVA засвідчують істотну різницю між середніми значеннями з показниками  $F = 13,56$ ;  $p = 1,94 \cdot 10^{-8}$ . Пост-хок аналіз Тьюкі (табл. 3.5) виявив значущі контрасти, зокрема  $COL > ABC$  на 34,2 м,  $Random > ABC$  на 37,71 м,  $COL > TOASL$  на 60,0 м та  $Random > TOASL$  на 63,5 м. Різниці  $ABC$ – $TOASL$  і  $COL$ – $Random$  не досягають статистичної значущості за  $\alpha = 0,05$ . У підсумку політики поділяються на дві групи ефективності. До групи з коротшими маршрутами належать ABC та TOASL, а Random і COL забезпечують довші маршрути.

Таблиця 3.5 – Пост-хок порівняння за критерієм Тьюкі з показниками пар груп, різниці середніх, скоригованого p, меж 95 % довірчого інтервалу та рішення щодо гіпотези

<b>group1</b>	<b>group2</b>	<b>meandiff</b>	<b>p-adj</b>	<b>lower</b>	<b>upper</b>	<b>reject</b>
ABC	COL	34,20	0,0174	4,33	64,05	True
ABC	Random	37,71	0,0067	7,85	67,57	True
ABC	TOASL	-25,79	0,1174	-55,65	4,07	False
COL	Random	3,52	0,9902	-26,34	33,38	False
COL	TOASL	-59,98	0,0000	-89,84	-30,12	True
Random	TOASL	-63,50	0,0000	-93,36	-33,64	True

Для інтерпретації практичного ефекту в табл. 3.4 наведено середні дистанції для замовлень на 5, 10, 15, 20 та 25 позицій. Кількість позицій збігається з кількістю місць відбору, які має відвідати комплектувальник, тобто кожна позиція відповідає одній комірці. Як видно з цієї таблиці, TOASL стабільно забезпечує найбільше скорочення шляху приблизно на 18–26 % у діапазоні від 5 до 25 позицій. ABC дає помірне, але стає зменшення на 9–18 %. Натомість COL здебільшого близький до Random або дещо гірший. За кількості 15 позицій перевищення становить близько 1 %. Така картина повністю корелює з медіанами та розкидом на рис. 3.9.

Отримані результати узгоджено показують, що поєднання кластерного відбору з часово-орієнтованим розміщенням TOASL мінімізує сумарну траєкторію руху. Підхід ABC також покращує показники порівняно з Random і COL, однак поступається TOASL і не відрізняється від нього статистично на рівні  $\alpha = 0,05$ . Візуальні індикатори на box plot (рис. 3.9) та числові оцінки у табл. 3.4 і 3.5 взаємно підкріплюють ці висновки та демонструють відтворюваність ефекту для різних розмірів партії.

### 3.9.2. mid-point: стратегія кластерного комплектування та порівняння політик розміщення

**Маршрут із серединою проходу (рис. 3.1, б).** За такого сценарію кожен прохід розглядається як поділений на дві половини відносно центральної лінії, яка може бути реальним поперечним коридором або умовною, програмно визначеною межею. Комплектувальник рухається вздовж коридорів і заходить лише в ті половини проходів, де є відбори. У кожній із них він прямує до найвіддаленішої від точки входу позиції відбору й повертається тим самим шляхом.

Порівняльний аналіз стратегій за маршрутом mid-point (рис. 3.10) свідчить про беззаперечну перевагу запропонованої політики TOASL, яка забезпечує найменші медіанні значення довжини шляху та стабільно вузький діапазон варіативності результатів. Політика ABC також демонструє кращу ефективність порівняно з методами Random та COL, для яких характерні значно вищі показники сумарної дистанції та широкий розкид значень, що вказує на їхню нестабільність у даній конфігурації. На графіку помітні **викиди (outliers)** в ABC та TOASL. Вони виникають у рідкісних конфігураціях, коли позиції замовлення розміщені у багатьох половинах проходів, ще й розосереджених між собою по різні боки. У таких випадках траєкторія містить багато заходів у різні половини та довші переходи між віддаленими зонами відбору вздовж коридорів доступу, що продовжує шлях обходу в окремих випадках, але не змінює загальну картину ефективності.

За даними вимірювань, на основі яких побудовано графік, середній пройдений шлях становить 218,6 м для Random, 169,3 м для ABC, 211,8 м для COL і 140,1 м для TOASL. Отже, TOASL забезпечує найкоротші маршрути. Політика ABC демонструє другий за ефективністю результат, поступаючись TOASL, проте систематично перевершує Random і COL. Наявні викиди відображають рідкісні конфігурації замовлень з віддаленим розташуванням точок відбору і не спотворюють узагальнений висновок.

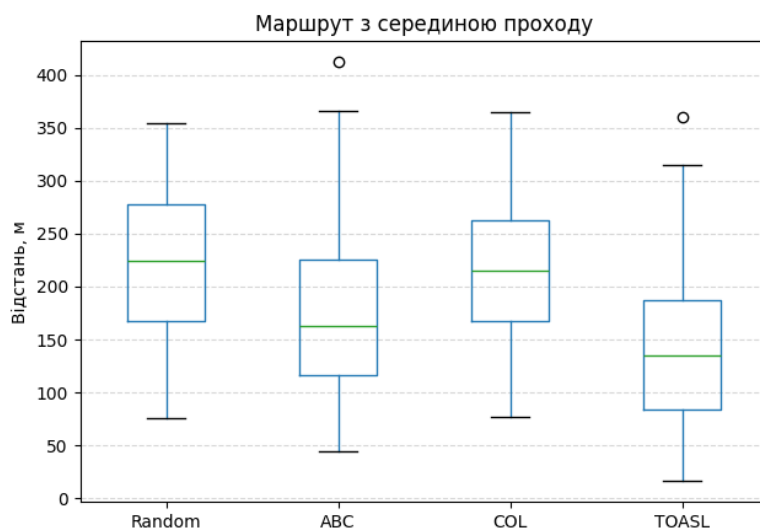


Рисунок 3.10 – Box plot довжин маршрутів для політик Random, ABC, COL і TOASL за маршрутом mid-point

Практичний ефект проілюстровано в табл. 3.6. Відносно Random політика TOASL стабільно скорочує довжину шляху приблизно на 52 %, 39 %, 37 %, 34 % і 29 % для партій із 5, 10, 15, 20 та 25 позицій. ABC забезпечує другий за ефективністю результат з типовим зменшенням на 21-30 % залежно від розміру партії. COL залишається близькою до Random зі змінами у межах 1–4 %.

Таблиця 3.6 – Середні дистанції (м) за різної кількості позицій для другого сценарію

К-сть позицій	Random	ABC	COL	TOASL
5	114,8	80,9 (–30 %)	113,4 (–1 %)	55,6 (–52 %)
10	182,5	144,6 (–21 %)	174,8 (–4 %)	111,7 (–39 %)
15	224,2	172,0 (–23 %)	215,7 (–4 %)	140,6 (–37 %)
20	271,0	210,6 (–22 %)	264,0 (–3 %)	178,2 (–34 %)
25	300,5	238,3 (–21 %)	291,2 (–3 %)	214,3 (–29 %)

Однофакторний дисперсійний аналіз підтверджує статистично значущі відмінності між середніми з такими показниками  $F = 28,896$ ;  $p = 6,47 \cdot 10^{-17}$ . Пост-хок тест Тьюкі (табл. 3.7) показує, що ABC має статистично меншу середню довжину маршруту, ніж COL і Random. У всіх порівняннях із TOASL

відмінності також значущі. Його середня дистанція є найменшою і нижчою за показники інших політик приблизно на 29–79 м (пари ABC–TOASL, COL–TOASL, Random–TOASL). Водночас COL і Random між собою статистично не відрізняються.

Таблиця 3.7 – Пост-хок порівняння за критерієм Тьюкі для другого сценарію

group1	group2	meandiff	p-adj	lower	upper	reject
ABC	COL	42,56	0,0001	17,43	67,70	True
ABC	Random	49,32	0,0000	24,19	74,45	True
ABC	TOASL	-29,18	0,0154	-54,31	-4,05	True
COL	Random	6,75	0,8997	-18,38	31,88	False
COL	TOASL	-71,74	0,0000	-96,87	-46,61	True
Random	TOASL	-78,50	0,0000	-103,63	-53,37	True

Часово-орієнтоване розміщення TOASL забезпечує найкоротші маршрути та має статистично підтверджену перевагу над ABC, COL і Random. Політика ABC систематично перевершує Random і COL, однак поступається TOASL. Отримані результати узгоджуються між собою на рівні графічного представлення (рис. 3.10), попарних порівнянь (табл. 3.7) та середніх дистанцій для різних розмірів партії (табл. 3.6).

### 3.9.3. largest gap: стратегія кластерного комплектування та порівняння політик розміщення

**Маршрут найбільшого розриву (рис. 3.1, в).** У кожному проході, де заплановано відбір, алгоритм визначає найдовший сегмент на якому таких точок немає. Рух здійснюється від ближчого коридору до найвіддаленішої позиції, далі – до межі цього сегмента. Після комплектувальник повертається тим самим шляхом, а відрізок, що відповідає найбільшому проміжку без точок, не проходиться. Половини проходів без відборів оминаються. Така логіка зменшує зайві переміщення в межах алеї й особливо ефективна, коли точки збирання розміщені поблизу одного з її кінців.



Статистична оцінка ефективності (рис. 3.11) підтверджує перевагу політики TOASL, яка забезпечує не лише мінімальну середню довжину маршруту, але й найвищу стабільність часу комплектування, що підтверджується вузьким діапазоном варіації. Стратегія ABC посідає друге місце за результативністю, суттєво випереджаючи методи Random та COL для яких характерні значні витрати на переміщення та великий розкид значень.

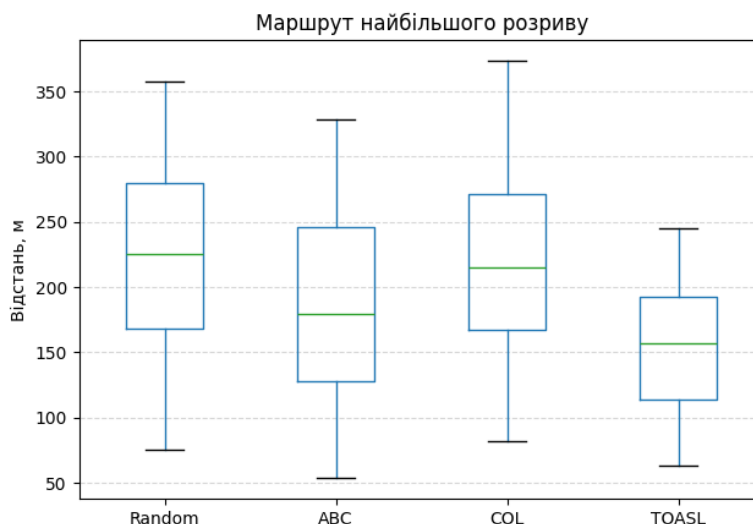


Рисунок 3.11 – Box plot довжин маршрутів для політик Random, ABC, COL і TOASL за маршрутом найбільшого розриву

За всією вибіркою середній шлях переміщення становить 221,96 м для Random, 185,00 м для ABC, 215,67 м для COL і 155,00 м для TOASL. За даними однофакторного ANOVA спостерігається значимий ефект впливу політики розміщення на середні показники  $F = 21,921$  та  $p = 3,75 \cdot 10^{-13}$ .

Практичний ефект ілюструє табл. 3.8. Відносно Random для TOASL спостерігається стабільне скорочення довжини шляху на 26 %, 24 %, 29 %, 31 % і 35 % для партій із 5, 10, 15, 20 та 25 позицій відповідно. ABC демонструє другий за ефективністю результат зі скороченням на 13–19 %, тоді як COL залишається близькою до Random з відхиленнями у межах 1–4 %. Така картина повністю узгоджується з розподілами на рис. 3.11.

Таблиця 3.8 – Середні дистанції (м) за різної кількості позицій для третього сценарію

К-сть позицій	Random	ABC	COL	TOASL
5	115,0	95,0 (-17 %)	113,7 (-1 %)	85,0 (-26 %)
10	183,8	160,0 (-13 %)	176,4 (-4 %)	140,0 (-24 %)
15	226,6	190,0 (-16 %)	219,2 (-3 %)	160,0 (-29 %)
20	276,5	230,0 (-17 %)	270,2 (-2 %)	190,0 (-31 %)
25	307,9	250,0 (-19 %)	298,8 (-3 %)	200,0 (-35 %)

**Пост-хок порівняння за критерієм Тьюкі.** Перевірка попарних контрастів показує, що методи чітко розшаровуються на дві групи: до групи коротких маршрутів належать TOASL і ABC, а до довгих – Random і COL (табл. 3.9). ABC має статистично менший середній пробіг ніж Random і COL. TOASL демонструє найкоротші траєкторії та істотно переважає всі інші, включно з ABC. Довірчі інтервали (95 % діапазони, у яких із заданою ймовірністю міститься істинна різниця середніх) повністю лежать у від’ємній області, що вказує на коротші маршрути. Пара COL–Random не виявляє значущих відмінностей за середньою довжиною шляху, тобто їхня ефективність подібна.

Таблиця 3.9 – Пост-хок порівняння за критерієм Тьюкі для третього сценарію

group1	group2	meandiff	p-adj	lower	upper	reject
ABC	COL	30,67	0,0059	6,64	54,70	True
ABC	Random	36,96	0,0005	12,93	60,99	True
ABC	TOASL	-30,00	0,0075	-54,03	-5,97	True
COL	Random	6,29	0,9065	-17,74	30,32	False
COL	TOASL	-60,67	0,0000	-84,70	-36,64	True
Random	TOASL	-66,96	0,0000	-90,99	-42,93	True

Поєднання кластерного відбору з часово-орієнтованим розміщенням TOASL виявляється найбільш оцадливим з точки зору логістичних витрат на переміщення та має статистично доведену перевагу над ABC, COL і Random. Політика ABC систематично перевершує Random і COL, але поступається

TOASL. Узгоджені результати отримано на рівні графічних індикаторів, середніх величин (табл. 3.8) і попарних тестів (табл. 3.9), що підтверджує відтворюваність ефекту для різних розмірів партій.

### 3.9.4. return: стратегія кластерного комплектування та порівняння політик розміщення

**Маршрут із поверненням (рис. 3.2, а).** За цього сценарію комплектувальник входить у прохід із боку найближчого коридору, рухається вздовж до найвіддаленішої точки збору і повертається тим самим шляхом. Ділянки проходів, у яких відповідні точки відсутні, оминає. Така логіка мінімізує зайві переміщення в межах алей, але загалом веде до довших пробігів, коли місця відбору розміщені далеко від коридорів доступу або зосереджені біля протилежних кінців проходу.

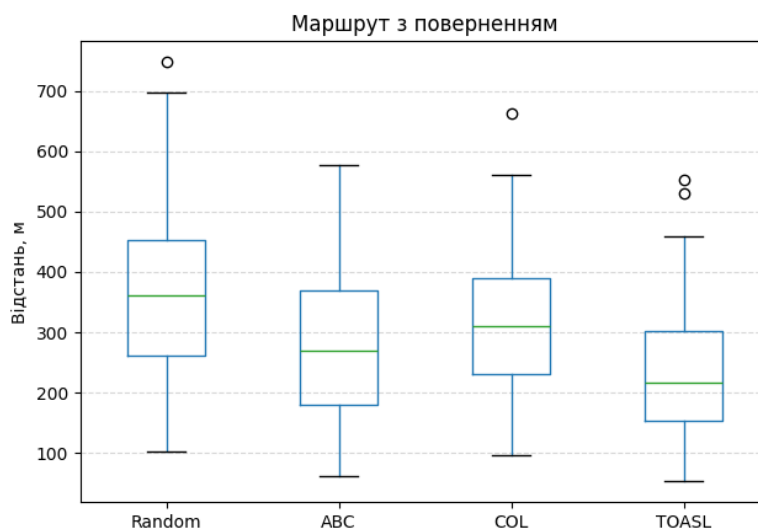


Рисунок 3.12 – Box plot довжин маршрутів для політик Random, ABC, COL і TOASL за маршрутом з поверненням

За діаграмою розмаху TOASL має найменші медіани та найвужчі міжквартильні діапазони, а ABC також демонструє нижчі медіани порівняно з COL і Random. У даних присутні поодинокі викиди. За правилом 1,5IQR поза «вусами» виявлено два спостереження для TOASL і по одному – для COL та

Random. Такі значення виникають у конфігураціях, коли точки збору розташовані дуже далеко від коридорів доступу або рознесені по протилежних кінцях проходів. У маршруті з поверненням це змушує комплектувальника далеко заходити в проходи та долати ті самі відстані у зворотному напрямку. Це коректні спостереження, що відображають геометрію маршруту, а не помилки вимірювань.

Як видно з табл. 3.10, зі зростанням кількості позицій абсолютні середні дистанції збільшуються для всіх політик, однак порядок їх ефективності не змінюється. TOASL забезпечує найкоротші маршрути за будь-якого розміру партії. Виграш відносно Random становить 31–37 %, найбільший ефект спостерігається для партій з 15 позицій. ABC стабільно посідає друге місце зі скороченням на 19–26 %, тоді як COL демонструє помірне зниження на 9–18 %. Таким чином, ранжування ефективності незмінне на всіх рівнях: TOASL → ABC → COL → Random.

Таблиця 3.10 – Середні дистанції (м) за різної кількості позицій для четвертого сценарію

К-сть позицій	Random	ABC	COL	TOASL
5	177,9	131,6 (–26 %)	161,3 (–9 %)	122,8 (–31 %)
10	295,8	241,0 (–19 %)	246,2 (–17 %)	195,4 (–34 %)
15	374,5	289,3 (–23 %)	307,7 (–18 %)	234,5 (–37 %)
20	462,8	348,7 (–25 %)	386,3 (–17 %)	302,5 (–35 %)
25	490,7	383,0 (–22 %)	440,5 (–10 %)	318,9 (–35 %)

**Підсумкові середні та ANOVA.** За всією вибіркою середній показник шляху становить 360,3 м для Random, 308,4 м для COL, 278,7 м для ABC і 234,8 м для TOASL. Проведений дисперсійний аналіз підтвердив статистичну розбіжність середніх значень із показниками  $F = 19,40$ ;  $p = 9,42 \cdot 10^{-12}$ .

**Пост-хок порівняння за критерієм Тьюкі (табл. 3.11).** Перевірка попарних контрастів виявляє чітке групування політик, причому до групи з коротшими дистанціями обходу належать TOASL та ABC, а до групи з

довшими траєкторіями руху – Random і COL. Зокрема, пост-хок контрасти показують, що TOASL забезпечує мінімальну довжину маршрутів і статистично переважає всі інші політики. Порівняно з ABC середня різниця становить  $-43,9$  м ( $p = 0,048$ ), причому 95-відсотковий довірчий інтервал цієї різниці повністю лежить у від’ємній області. Відносно COL різниця дорівнює  $-73,6$  м ( $p = 0,0001$ ), а відносно Random –  $-125,5$  м ( $p < 0,0001$ ). Політика ABC, своєю чергою, істотно краща за Random ( $-81,6$  м;  $p < 0,0001$ ), тоді як відмінностей між ABC і COL не виявлено ( $p = 0,298$ ). Нарешті, COL також статистично скорочує шлях порівняно з Random на  $-51,9$  м ( $p = 0,012$ ).

Таблиця 3.11 – Пост-хок порівняння за критерієм Тьюкі для четвертого сценарію

group1	group2	meandiff	p-adj	lower	upper	reject
ABC	COL	29,67	0,2977	-13,99	73,33	FALSE
ABC	Random	81,59	0,0000	37,94	125,25	TRUE
ABC	TOASL	-43,90	0,0481	-87,56	-0,24	TRUE
COL	Random	51,93	0,0123	8,27	95,59	TRUE
COL	TOASL	-73,57	0,0001	-117,23	-29,91	TRUE
Random	TOASL	-125,50	0,0000	-169,16	-81,84	TRUE

Поєднання кластерного відбору з часово орієнтованим розміщенням TOASL показує найкращий результат щодо оптимізації шляху для цієї конфігурації. ABC демонструє другий за ефективністю результат. Різниця між COL і Random невелика, а відмінність між ABC та COL статистично не підтверджується. Загальна картина на рис. 3.12 узгоджується з числовими оцінками у табл. 3.10 та табл. 3.11 і підкреслює відтворюваність ефекту для різних розмірів партій.

### 3.9.5. composite: стратегія кластерного комплектування та порівняння політик розміщення

Стратегія комбінованого маршруту (composite) поєднує в собі елементи повного проходження рядів і повернення, обираючи варіант для кожного

проходу залежно від того, який з них забезпечує меншу додаткову відстань (рис. 3.2, б). Таким чином, у межах одного замовлення комплектувальник може здійснювати повернення у тих проходах, де всі необхідні позиції знаходяться близько до входу, і водночас проходити наскрізь ряди з товарами в глибині складу. Такий підхід дозволяє адаптивно скорочувати маршрути й, на відміну від більш жорстких політик, стабільно демонструє кращі результати за різних конфігурацій розміщення товарів.

На основі порівняльного аналізу пройдених відстаней (рис. 3.13) встановлено, що політика TOASL є найефективнішою стратегією для комбінованого маршруту. Дослідження варіативності вказує на те, що застосування TOASL гарантує короткі маршрути незалежно від складу замовлення. Натомість найгірші показники характерні для політики COL, яка в даній конфігурації призводить до надмірних переміщень, тоді як метод Random займає проміжну позицію, демонструючи дещо кращі результати за COL, але значно поступаючись оптимізованим підходам.

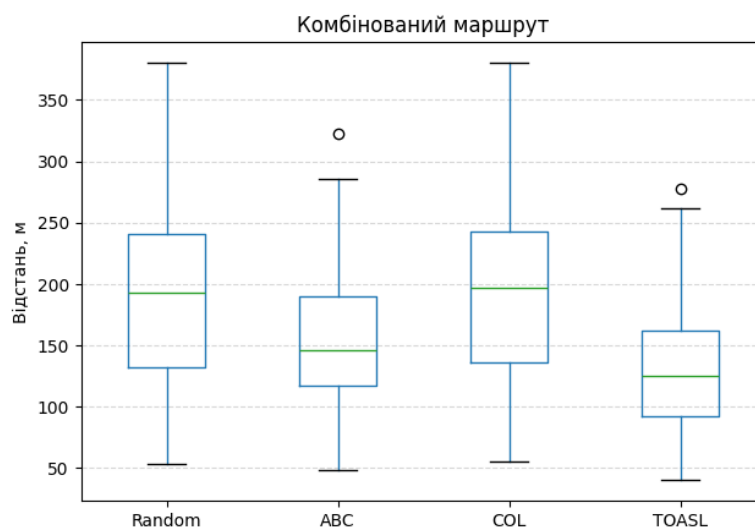


Рисунок 3.13 – Box plot довжин маршрутів для політик Random, ABC, COL і TOASL за комбінованим маршрутом

У табл. 3.12 представлено середні значення довжин маршрутів для замовлень із різною кількістю позицій. Для прикладу, при 5 позиціях середня відстань становила 98,6 м для Random, тоді як застосування ABC знижує її на

26 % до 72,7 м, а TOASL на 44 % до 55,2 м. Для більших замовлень ефективність TOASL зберігається, демонструючи стабільне зменшення шляху на 32–44 % порівняно з випадковим розміщенням. COL, навпаки, у більшості випадків формує довші маршрути через нерівномірний просторовий розподіл SKU відносно точки I/O, оскільки значна частина затребуваних позицій опиняється у глибоких відсіках проходів, що збільшує довжину обходів.

Таблиця 3.12 – Середні дистанції (м) за різної кількості позицій для п'ятого сценарію

К-сть позицій	Random	ABC	COL	TOASL
5	98,6	72,7 (–26 %)	99,9 (+1 %)	55,2 (–44 %)
10	159,5	133,4 (–16 %)	167,6 (+5 %)	108,5 (–32 %)
15	197,3	147,5 (–25 %)	205,7 (+4 %)	124,1 (–37 %)
20	238,8	187,5 (–21 %)	235,7 (–1 %)	159,2 (–33 %)
25	274,6	220,8 (–20 %)	278,9 (+2 %)	187,4 (–32 %)

Для перевірки статистичної значущості різниці між політиками розміщення було застосовано пост-хок тест Тьюкі. Результати табл. 3.13 свідчать, що різниця між Random і ABC, а також між Random і TOASL є статистично значущою при рівні  $p < 0,001$ . При цьому порівняння між COL і Random показало відсутність істотних відмінностей. Це підтверджує, що TOASL та ABC дійсно забезпечують кращу ефективність при комбінованому маршруті, тоді як COL не має переваг.

Таблиця 3.13 – Пост-хок порівняння за критерієм Тьюкі для п'ятого сценарію

group1	group2	meandiff	p_adj	lower	upper	reject
ABC	COL	45,21	0,0000	21,38	69,04	True
ABC	Random	41,38	0,0001	17,55	65,22	True
ABC	TOASL	-25,45	0,0311	-49,29	-1,62	True
COL	Random	-3,83	0,9760	-27,66	20,01	False
COL	TOASL	-70,66	0,0000	-94,50	-46,83	True
Random	TOASL	-66,84	0,0000	-90,67	-43,00	True

Аналіз результатів моделювання підтверджує, що комбінований маршрут у поєднанні з оптимізованими політиками розміщення, зокрема TOASL, забезпечує найкоротші траєкторії руху. На відміну від більш жорстких стратегій маршрутизації, composite виявляє високу адаптивність і забезпечує скорочення сумарних витрат на переміщення в межах складу. Таким чином, TOASL у взаємодії з комбінованим маршрутом можна вважати найбільш перспективним підходом, що поєднує гнучкість маршрутів і ефективність політик розміщення.

### **Висновки до розділу 3**

У розглянутому розділі проаналізовано задачу оптимального розміщення товарів (SLAP) з урахуванням маршрутизації комплектувальника та довжин маршрутів на складі, а також запропоновано комплексний підхід до її модельної верифікації. Розроблено методику TOASL в основі якої лежать передові алгоритмічні рішення кластеризації часових рядів і поетапна архітектура. Етап підготовки даних передбачає очищення, нормалізацію й агрегування часових рядів попиту. На етапі моделювання формуються кластери/зони та правила розміщення SKU. Під час симуляції відтворюються сценарії роботи складу й маршрути комплектувальника для різних політик. Етап оцінювання забезпечує зіставлення політик за ключовими метриками, зокрема за довжиною маршруту, а також статистичну перевірку відмінностей. Розроблена методика TOASL інтегрує етапи розміщення та комплектування товарів в єдину оптимізаційну логіку, уникаючи однобічної орієнтації лише на відбиранні замовлень.

Розроблено алгоритм зонування складу на основі кластеризації попиту. Дані згруповано у шістнадцять кластерів і виконано внутрішньозонове впорядкування SKU. Далі проведено аналіз і обрано політики маршрутизації S-shape, mid-point, largest gap, return та composite, а також визначено сценарії комплектування. Розроблена методика являє собою відтворюваний цикл



експериментів і чіткі правила обчислення відстаней, що забезпечує порівнюваність результатів між схемами Random, ABC, COL та TOASL.

Для доведення ефективності здійснено комплексне моделювання. Експериментальний дизайн передбачав десять поставок, чотири політики розміщення, п'ять маршрутів комплектування та п'ять рівнів розміру партії замовлень. Для кожної комбінації факторів виконано по два незалежні повторення, що разом становить 2000 спостережень.

Статистична обробка охоплювала однофакторний дисперсійний аналіз (ANOVA) окремо для кожного сценарію. Для виявлення парних відмінностей застосовано пост-хок порівняння за критерієм Tukey HSD з обчисленням 95 % довірчих інтервалів ефектів, що забезпечує статистично обґрунтовані та відтворювані висновки.

Часово-орієнтоване розміщення TOASL забезпечує найкоротші маршрути та демонструє доведену перевагу над ABC, COL і Random для низки сценаріїв. Наприклад, за комбінованого маршруту TOASL скорочує середню довжину шляху на 32–44 % порівняно з Random і на 15–24 % порівняно з ABC, причому ефект зберігається для різних розмірів партії. При цьому різниці між TOASL і політиками COL та Random залишаються статистично значущими, тоді як COL не виявляє переваги над Random.

Застосування методики TOASL може суттєво підвищити ефективність роботи складу завдяки поєднанню потоків укладання на зберігання та збору товарів і використанню перевірених експериментами правил розміщення.

## ВИСНОВКИ

У дисертаційній роботі розв'язано актуальне науково-прикладне завдання оптимізації обробки інформації в логістичних центрах шляхом розробки та впровадження методів інтелектуального аналізу даних на основі часових рядів.

Узагальнення результатів проведеного дослідження дозволяє виділити такі найбільш вагомні наукові та практичні здобутки:

1. Проаналізовано стан методів та алгоритмів управління інформаційними потоками в логістиці в умовах Індустрії 4.0. Доведено обмеженість підходів ABC, XYZ та VEN через ігнорування часової структури попиту, що обґрунтовує необхідність інтелектуалізації WMS-систем для роботи в динамічному середовищі.

2. Доведено доцільність використання часових рядів попиту для сегментації товарів, де джерелом вхідної інформації слугують дані з систем класу ERP, WMS та CRM. Застосування метрик DTW та SBD дозволяє виявляти приховані патерни замовлень за формою попиту, що підвищує точність групування порівняно з класичними методами.

3. Розроблено методику Часово-орієнтованого Призначення Місць Зберігання (TOASL). Запропонований підхід інтегрує етапи підготовки даних, кластеризації, зонування складу та вибору політик маршрутизації в єдину оптимізаційну логіку. Це дозволяє уникнути однобічної орієнтації лише на відбиранні замовлень, поєднуючи потоки укладання на зберігання та збору товарів.

4. Експериментально підтверджено переваги запропонованого підходу. Моделювання на основі 2000 спостережень та аналіз за методами ANOVA і Tukey HSD довели, що застосування TOASL дозволяє досягти суттєвого зменшення довжини маршрутів. Наприклад, при розміщенні товарів за політикою TOASL та за комбінованої стратегії збирання досягнуто скорочення дистанцій збору на 32% порівняно з Random та на 15% відносно

ABC, причому висока ефективність зберігається для різних розмірів партій замовлень.

5. Оцінено вплив на процеси розміщення. Встановлено, що за використання TOASL дистанція укладання на зберігання товарів збільшується лише на 5,3 % порівняно з Random, що є цілком прогнозованим результатом. З огляду на суттєве скорочення маршрутів комплектування, TOASL є раціональним рішенням, що поєднує прийнятні витрати на зберігання з високою ефективністю виконання замовлень.

Результати досліджень можуть бути використані розробниками WMS-систем та логістичними компаніями для підвищення пропускної здатності складів. Застосування методики TOASL дозволяє логістичним центрам суттєво скоротити операційні витрати шляхом мінімізації переміщень складської техніки та часу роботи персоналу. Ба більше, впровадження розроблених алгоритмів сприяє автоматизації процесів зонування складських площ, що забезпечує гнучку адаптацію до сезонних коливань попиту.

Дисертаційна робота була виконана в рамках госпдоговірної науково-дослідної роботи «Комплексна розробка прикладних IT-рішень для підвищення продуктивності комп'ютерних систем у комерційному та соціальному секторі» (Державний реєстраційний номер 0125U003178), Державного університету інформаційно-комунікаційних технологій.

Перспективи майбутніх досліджень полягають у подальшому вдосконаленні та розвитку гібридних стратегій розміщення. Зокрема, пріоритетним напрямом є адаптація методики TOASL до роботизованих складських систем. Оптимізація маршрутів роботизованих агентів на основі запропонованого підходу дозволить забезпечити високу ефективність системи, оскільки скорочення дистанцій переміщення прямо корелює зі зменшенням енергоспоживання, зниженням зносу механічних вузлів та продовженням терміну служби акумуляторних батарей.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Лемешко А. В., Балвак А. ОПТИМІЗАЦІЯ ОБРОБЛЕННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ СУЧАСНИХ КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ У РОБОТІ ЛОГІСТИЧНИХ ЦЕНТРІВ. *ЗВ'ЯЗОК*. 2022. № 2. С. 66–69. URL: <https://doi.org/10.31673/2412-9070.2022.026669>
2. АКТУАЛЬНІ ЗАСАДИ СТВОРЕННЯ АЛГОРИТМІВ ОБРОБКИ ІНФОРМАЦІЇ ДЛЯ ЛОГІСТИЧНИХ ЦЕНТРІВ / А. В. Лемешко та ін. *Таврійський науковий вісник. Серія: Технічні науки*. 2023. № 1. С. 25–32. URL: <https://doi.org/10.32851/tnv-tech.2023.1.3>
3. ОБРОБКА ТА АНАЛІЗ ДАНИХ НА ПРИКЛАДІ НАБОРУ SPAMBASE З ВИКОРИСТАННЯМ БІБЛІОТЕК ДЛЯ МАШИННОГО НАВЧАННЯ / А. А. Балвак та ін. *Таврійський науковий вісник. Серія: Технічні науки*. 2024. № 2. С. 3–20. URL: <https://doi.org/10.32782/tnv-tech.2024.2.1>
4. Балвак А., Лащевська Н. ІНТЕЛЕКТУАЛЬНІ ПІДХОДИ ДО ОПТИМІЗАЦІЇ СКЛАДСЬКИХ ПРОЦЕСІВ У ЗАДАЧАХ РОЗМІЩЕННЯ КОМПЛЕКТУВАННЯ ТА РОБОТИЗАЦІЇ. *Кібербезпека: освіта, наука, техніка*. 2025. Т. 1, № 29. С. 161–177. URL: <https://doi.org/10.28925/2663-4023.2025.29.869>
5. Балвак А., Лащевська Н. МЕТОДИКА ЗБАЛАНСОВАНОГО ПРИЗНАЧЕННЯ МІСЦЬ ЗБЕРІГАННЯ ТОВАРІВ НА ОСНОВІ ЧАСОВИХ ХАРАКТЕРИСТИК ПОПИТУ. *Кібербезпека: освіта, наука, техніка*. 2025 Т. 3, № 31. С. 155–168. URL: <https://doi.org/10.28925/2663-4023.2025.31.1002>
6. Балвак А.А. Зінченко В.В. ФОРМУВАННЯ ВИБІРКИ ДАНИХ ТА КРИТЕРІЇ ЯКОСТІ ДЛЯ МОДЕЛЮВАННЯ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ СКЛАДСЬКОЇ ЛОГІСТИКИ. *ЗВ'ЯЗОК*. 2026. № 1. С. 96-104. <https://doi.org/10.31673/2412-9070.2026.017401>
7. Integrated Modeling in the Quality Assessment of Flight Management Software Systems / О. Zinchenko та ін. *Informatica*. 2025. Т. 49, № 31. URL: <https://doi.org/10.31449/inf.v49i31.11012>

8. de Koster R., Le-Duc T., Roodbergen K. J. Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*. 2007. T. 182, № 2. C. 481–501. URL: <https://doi.org/10.1016/j.ejor.2006.07.009>.
9. Azizpour M., Namazypour N., Kirchheim A. Synthetic Data Generation for Robotic Order Picking. *Logistics Journal: Proceedings*. T. 18. URL: [https://doi.org/10.2195/lj\\_proc\\_azizpour\\_en\\_202211\\_01](https://doi.org/10.2195/lj_proc_azizpour_en_202211_01).
10. Pang K.-W., Chan H.-L. Data mining based algorithm for storage location assignment in a randomised warehouse. *International Journal of Production Research*. 2016. T. 55, № 14. C. 4035–4052. URL: <https://doi.org/10.1080/00207543.2016.1244615>.
11. Allgor R., Cezik T., Chen D. Algorithm for Robotic Picking in Amazon Fulfillment Centers Enables Humans and Robots to Work Together Effectively. *INFORMS Journal on Applied Analytics*. 2023. URL: <https://doi.org/10.1287/inte.2022.1143>.
12. Chen H., Wang Z., Zhang M. Mechanism Study on Innovative Influences of AI and Blockchain on Supply Chain Logistics: Case Study of JD and Alibaba. *Advances in Economics, Management and Political Sciences*. 2024. T. 109, № 1. C. 94–100. URL: <https://doi.org/10.54254/2754-1169/109/2024bj0116>.
13. Winkelhaus S., Grosse E. H. Logistics 4.0: a systematic review towards a new logistics system. *International Journal of Production Research*. 2019. T. 58, № 1. C. 18–43. URL: <https://doi.org/10.1080/00207543.2019.1612964>.
14. Binos T., Bruno V., Adamopoulos A. Intelligent agent based framework to augment warehouse management systems for dynamic demand environments. *Australasian Journal of Information Systems*. 2021. T. 25. URL: <https://doi.org/10.3127/ajis.v25i0.2845>.
15. Jemelka M., Chramcov B., Kříž P. Design of the storage location based on the ABC analyses. *INTERNATIONAL CONFERENCE OF NUMERICAL ANALYSIS AND APPLIED MATHEMATICS 2015 (ICNAAM 2015)*, M. Rhodes, Greece. 2016. URL: <https://doi.org/10.1063/1.4951909>.

16. Integration of demand forecasts in ABC-XYZ analysis: practical investigation at an industrial company / B. Scholz-Reiter та ін. *International Journal of Productivity and Performance Management*. 2012. Т. 61, № 4. С. 445–451. URL: <https://doi.org/10.1108/17410401211212689>.

17. Inayah I., Alexandri B., Pragiwani M. Economic Order Quantity (EOQ) Method Analysis, ABC Classification and Vital, Essential and Non Essential (VEN) Analysis of Medicines. *Indonesian Journal of Business, Accounting and Management*. 2022. Т. 5, № 1. С. 15–24. URL: <https://doi.org/10.36406/ijbam.v5i01.562>.

18. Designing efficient order picking systems by combining planning problems: State-of-the-art classification and review / T. van Gils та ін. *European Journal of Operational Research*. 2018. Т. 267, № 1. С. 1–15. URL: <https://doi.org/10.1016/j.ejor.2017.09.002>.

19. Masae M., Glock C. H., Grosse E. H. Order picker routing in warehouses: A systematic literature review. *International Journal of Production Economics*. 2020. Т. 224. С. 107564. URL: <https://doi.org/10.1016/j.ijpe.2019.107564>.

20. Optimization of Warehouse Operations with Genetic Algorithms / M. Kordos та ін. *Applied Sciences*. 2020. Т. 10, № 14. С. 4817. URL: <https://doi.org/10.3390/app10144817>.

21. A Novel Parallel Ant Colony Optimization Algorithm for Warehouse Path Planning / J. Yu та ін. *Journal of Control Science and Engineering*. 2020. Т. 2020. С. 1–12. URL: <https://doi.org/10.1155/2020/5287189>.

22. Reyes J. J. R., Solano-Charris E. L., Montoya-Torres J. R. The storage location assignment problem: A literature review. *International Journal of Industrial Engineering Computations*. 2019. С. 199–224. URL: <https://doi.org/10.5267/j.ijiec.2018.8.001>.

23. Dynamic Storage Location Assignment in Warehouses Using Deep Reinforcement Learning / C. Waubert de Puiseau та ін. *Technologies*. 2022. Т. 10, № 6. С. 129. URL: <https://doi.org/10.3390/technologies10060129>.

24. Adaptive warehouse storage location assignment with considerations to order-picking efficiency and worker safety / A. Zarinchang та ін. *Journal of Industrial and Production Engineering*. 2023. C. 1–20. URL: <https://doi.org/10.1080/21681015.2023.2263009>.
25. Tejesh B. S. S., Neeraja S. Warehouse inventory management system using IoT and open source framework. *Alexandria Engineering Journal*. 2018. T. 57, № 4. C. 3817–3823. URL: <https://doi.org/10.1016/j.aej.2018.02.003>.
26. Singh R., Singh E., Nalwa H. S. Inkjet printed nanomaterial based flexible radio frequency identification (RFID) tag sensors for the internet of nano things. *RSC Adv*. 2017. T. 7, № 77. C. 48597–48630. URL: <https://doi.org/10.1039/c7ra07191d>.
27. Zhu L., Wang P., Xi S. Mean-Variance Analysis of Retailers Deploying RFID-Enabled Smart Shelves. *Information*. 2018. T. 9, № 2. C. 40. URL: <https://doi.org/10.3390/info9020040>.
28. Uppalapati P. K. S. Optimizing Warehouse Operations through Real-Time Integration with Internet of Things Devices. *European Modern Studies Journal*. 2025. T. 9, № 3. C. 483–491. URL: [https://doi.org/10.59573/emsj.9\(3\).2025.43](https://doi.org/10.59573/emsj.9(3).2025.43).
29. A Novel Parts-to-Picker System with Buffer Racks and Access Racks in Flexible Warehousing Systems / M. He та ін. *Sustainability*. 2024. T. 16, № 4. C. 1388. URL: <https://doi.org/10.3390/su16041388>.
30. Pan Q. Design and analysis of an autonomous warehouse robot system with 6-DOF manipulator. *Applied and Computational Engineering*. 2024. T. 34, № 1. C. 114–121. URL: <https://doi.org/10.54254/2755-2721/34/20230310>.
31. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications / B. T. Hazen та ін. *International Journal of Production Economics*. 2014. T. 154. C. 72–80. URL: <https://doi.org/10.1016/j.ijpe.2014.04.018>.
32. Zhou H., Ning W., Guo T. A fractal deep learning model for optimizing e-commerce warehousing and logistics. *Journal of Industrial and Management*

*Optimization.* 2026. T. 22, № 1. C. 374–404.

URL: <https://doi.org/10.3934/jimo.2026014>.

33. Wang M., Zhang R.-Q., Fan K. Improving order-picking operation through efficient storage location assignment: A new approach. *Computers & Industrial Engineering.* 2020. T. 139. C. 106186. URL: <https://doi.org/10.1016/j.cie.2019.106186>.

34. Application of Artificial intelligence in Logistics 4.0: DHL case study analysis / B. Boorová та ін. *Ekonomika preduzeca.* 2024. T. 72, № 5-6. C. 292–304. URL: <https://doi.org/10.5937/ekopre2406292b>.

35. Brightpick Launches Autopicker 2.0, the First Multi-Purpose Warehouse Robot to Match Human-Level Performance - Brightpick. *Brightpick.* URL: <https://brightpick.ai/brightpick-launches-autopicker-2>.

36. Fatima S. S. W., Rahimi A. A Review of Time-Series Forecasting Algorithms for Industrial Manufacturing Systems. *Machines.* 2024. T. 12, № 6. C. 380. URL: <https://doi.org/10.3390/machines12060380>.

37. Martins E., Galegale N. V. Sales forecasting using machine learning algorithms. *Revista de Gestão e Secretariado (Management and Administrative Professional Review).* 2023. T. 14, № 7. C. 11294–11308. URL: <https://doi.org/10.7769/gesec.v14i7.1670>.

38. Babai M. Z., Boylan J. E., Rostami-Tabar B. Demand forecasting in supply chains: a review of aggregation and hierarchical approaches. *International Journal of Production Research.* 2021. T. 60, № 1. C. 324–348. URL: <https://doi.org/10.1080/00207543.2021.2005268>.

39. Forecasting hierarchical time series in supply chains: an empirical investigation / D. Mircetic та ін. *International Journal of Production Research.* 2021. C. 1–20. URL: <https://doi.org/10.1080/00207543.2021.1896817>.

40. An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain / Z. H. Kilimci та ін. *Complexity.* 2019. T. 2019. C. 1–15. URL: <https://doi.org/10.1155/2019/9067367>.



41. Demand Forecasting Tool For Inventory Control Smart Systems / F. Z. Benhamida та ін. *Journal of Communications Software and Systems*. 2021. Т. 17, № 2. С. 185–196. URL: <https://doi.org/10.24138/jcomss-2021-0068>.
42. An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis / K. Nikolopoulos та ін. *Journal of the Operational Research Society*. 2011. Т. 62, № 3. С. 544–554. URL: <https://doi.org/10.1016/j.jors.2010.32>.
43. van Ruitenbeek R. E., Koole G. M., Bhulai S. A hierarchical agglomerative clustering for product sales forecasting. *Decision Analytics Journal*. 2023. С. 100318. URL: <https://doi.org/10.1016/j.dajour.2023.100318>.
44. Rahman M. A., Sarker B. R. A Bayesian approach to forecast intermittent demand for seasonal products. *International Journal of Industrial and Systems Engineering*. 2012. Т. 11, № 1/2. С. 137. URL: <https://doi.org/10.1504/ijise.2012.046660>.
45. Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion / M. Abolghasemi та ін. *Computers & Industrial Engineering*. 2020. Т. 142. С. 106380. URL: <https://doi.org/10.1016/j.cie.2020.106380>.
46. Yadav S. A Comparative Study of ARIMA, Prophet and LSTM for Time Series Prediction. *Journal of Artificial Intelligence, Machine Learning and Data Science*. 2023. Т. 1, № 1. С. 1813–1816. URL: <https://doi.org/10.51219/jaimld/sandeep-yadav/402>.
47. Taylan A., Yapar G., Selamlar H. Automatic Time Series Forecasting with Ata Method in R: ATAForecasting Package. *The R Journal*. 2021. Т. 13, № 2. С. 441. URL: <https://doi.org/10.32614/rj-2021-101>.
48. Hu C., Paunic V. Building Forecasting Solutions Using Open-Source and Azure Machine Learning. *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, м. Virtual Event CA USA. New York, NY, USA, 2020. URL: <https://doi.org/10.1145/3394486.3406699>.

49. Granillo-Macías R. Inventory management and logistics optimization: A data mining practical approach. *Logforum*. 2020. T. 16, № 4. C. 535–547. URL: <https://doi.org/10.17270/j.log.2020.512>.

50. Pujiarto B. A Data Mining Practical Approach to Inventory Management and Logistics Optimization. *IJIIS: International Journal of Informatics and Information Systems*. 2021. T. 4, № 2. C. 112–122. URL: <https://doi.org/10.47738/ijiis.v4i2.109>.

51. Optimizing Clustering Techniques for Retail Industry: One-Dimensional Time Warping Method for Demand Uncertainty / M. N. P. Ma'ady та ін. *SIET 2023: International Conference on Sustainable Information Engineering and Technology*, м. Badung, Bali Indonesia. New York, NY, USA, 2023. URL: <https://doi.org/10.1145/3626641.3627602>.

52. Anh D. T., Thanh L. H. An efficient implementation of k-means clustering for time series data with DTW distance. *International Journal of Business Intelligence and Data Mining*. 2015. T. 10, № 3. C. 213. URL: <https://doi.org/10.1504/ijbidm.2015.071311>.

53. Łuczak M. Hierarchical clustering of time series data with parametric derivative dynamic time warping. *Expert Systems with Applications*. 2016. T. 62. C. 116–130. URL: <https://doi.org/10.1016/j.eswa.2016.06.012>.

54. Paparrizos J., Gravano L. k-Shape. *SIGMOD/PODS'15: International Conference on Management of Data*, м. Melbourne Victoria Australia. New York, NY, USA, 2015.

55. Paparrizos J., Gravano L. Fast and Accurate Time-Series Clustering. *ACM Transactions on Database Systems*. 2017. T. 42, № 2. C. 1–49. URL: <https://doi.org/10.1145/3044711>.

56. Delineating urban functional areas with building-level social media data: A dynamic time warping (DTW) distance based k-medoids method / Y. Chen та ін. *Landscape and Urban Planning*. 2017. T. 160. C. 48–60. URL: <https://doi.org/10.1016/j.landurbplan.2016.12.001>.

57. Holder C., Guijo-Rubio D., Bagnall A. Clustering Time Series with k-Medoids Based Algorithms. *Advanced Analytics and Learning on Temporal Data*. Cham, 2023. С. 39–55. URL: [https://doi.org/10.1007/978-3-031-49896-1\\_4](https://doi.org/10.1007/978-3-031-49896-1_4).
58. Cuturi, Marco & Blondel, Mathieu. (2017). Soft-DTW: a Differentiable Loss Function for Time-Series. 10.48550/arXiv.1703.01541.
59. Huy V. T., Anh D. T. An efficient implementation of anytime k-medoids clustering for time series under dynamic time warping. *SoICT '16: Seventh International Symposium on Information and Communication Technology*, м. Ho Chi Minh City Vietnam. New York, NY, USA, 2016. URL: <https://doi.org/10.1145/3011077.3011128>.
60. Temporal Customer Segmentation Using the Self-organizing Time Map / Z. Yao та ін. *2012 16th International Conference on Information Visualisation (IV)*, м. Montpellier, France, 11–13 лип. 2012 р. URL: <https://doi.org/10.1109/iv.2012.47>.
61. Theoretical and Applied Aspects of the Self-Organizing Maps / M. Cottrell та ін. *Advances in Self-Organizing Maps and Learning Vector Quantization*. Cham, 2016. С. 3–26. URL: [https://doi.org/10.1007/978-3-319-28518-4\\_1](https://doi.org/10.1007/978-3-319-28518-4_1).
62. Kondruk N. E. A COMPARATIVE STUDY OF CLUSTER VALIDITY INDICES. *Radio Electronics, Computer Science, Control*. 2019. № 4. С. 59–67. URL: <https://doi.org/10.15588/1607-3274-2019-4-6>.
63. Improved the Performance of the K-Means Cluster Using the Sum of Squared Error (SSE) optimized by using the Elbow Method / R. Nainggolan та ін. *Journal of Physics: Conference Series*. 2019. Т. 1361. С. 012015. URL: <https://doi.org/10.1088/1742-6596/1361/1/012015>.
64. Umargono E., Suseno J. E., S. K. V. G. K-Means Clustering Optimization using the Elbow Method and Early Centroid Determination Based-on Mean and Median. *International Conferences on Information System and Technology*, м. Yogyakarta, Indonesia, 5–6 груд. 2019 р. URL: <https://doi.org/10.5220/0009908402340240>.

65. Gustriansyah R., Suhandi N., Antony F. Clustering optimization in RFM analysis Based on k-Means. *Indonesian Journal of Electrical Engineering and Computer Science*. 2020. T. 18, № 1. C. 470. URL: <https://doi.org/10.11591/ijeecs.v18.i1.pp470-477>.

66. Mehar A. M., Matawie K., Maeder A. Determining an optimal value of K in K-means clustering. *2013 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, м. Shanghai, China, 18–21 груд. 2013 р. URL: <https://doi.org/10.1109/bibm.2013.6732734>.

67. Novel clustering selection criterion for fast binary key speaker diarization / H. Delgado та ін. *Interspeech* 2015. ISCA, 2015. URL: <https://doi.org/10.21437/interspeech.2015-108>.

68. Enhancing Warehouse Efficiency with Time Series Clustering: A Hybrid Storage Location Assignment Strategy / H. Kalkha та ін. *IEEE Access*. 2024. C. 1. URL: <https://doi.org/10.1109/access.2024.3386887>.

69. Tibshirani R., Walther G., Hastie T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2001. T. 63, № 2. C. 411–423. URL: <https://doi.org/10.1111/1467-9868.00293>.

70. Review-based method for evaluating key performance indicators: an application on warehouse system / A. Faveto та ін. *The International Journal of Advanced Manufacturing Technology*. 2023. URL: <https://doi.org/10.1007/s00170-023-12684-4>.

71. Umargono E., Suseno J. E., Vincensius Gunawan S. K. K-Means Clustering Optimization Using the Elbow Method and Early Centroid Determination Based on Mean and Median Formula. *The 2nd International Seminar on Science and Technology (ISSTEC 2019)*, м. Yogyakarta, Indonesia, 25–26 листоп. 2019 р. Paris, France, 2020. URL: <https://doi.org/10.2991/assehr.k.201010.019>.

72. Shahapure K. R., Nicholas C. Cluster Quality Analysis Using Silhouette Score. *2020 IEEE 7th International Conference on Data Science and Advanced*

*Analytics (DSAA)*, м. Sydney, Australia, 6–9 жовт. 2020 р. 2020.  
URL: <https://doi.org/10.1109/dsaa49011.2020.00096>.

73. Warehouse design and control: Framework and literature review / B. Rouwenhorst та ін. *European Journal of Operational Research*. 2000. Т. 122, № 3. С. 515–533. URL: [https://doi.org/10.1016/s0377-2217\(99\)00020-x](https://doi.org/10.1016/s0377-2217(99)00020-x).

74. Hausman W. H., Schwarz L. B., Graves S. C. Optimal Storage Assignment in Automatic Warehousing Systems. *Management Science*. 1976. Т. 22, № 6. С. 629–638. URL: <https://doi.org/10.1287/mnsc.22.6.629>.

75. Petersen C. G., Aase G. A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*. 2004. Т. 92, № 1. С. 11–19. URL: <https://doi.org/10.1016/j.ijpe.2003.09.006>.

76. HALL R. W. DISTANCE APPROXIMATIONS FOR ROUTING MANUAL PICKERS IN A WAREHOUSE. *IIE Transactions*. 1993. Т. 25, № 4. С. 76–87. URL: <https://doi.org/10.1080/07408179308964306>.

77. Кучерук О. Я., Драч І. В. ОПТИМІЗАЦІЙНИЙ ПІДХІД В ЗАДАЧІ МАРШРУТИЗАЦІЇ КОМПЛЕКТУВАЛЬНИКА. *Вісник Черкаського державного технологічного університету*. 2021. № 3. С. 59–68. URL: <https://doi.org/10.24025/2306-4412.3.2021.240748>.

78. Petersen C. G. An evaluation of order picking routeing policies. *International Journal of Operations & Production Management*. 1997. Т. 17, № 11. С. 1098–1111. URL: <https://doi.org/10.1108/01443579710177860>.

79. Roodbergen K. J., de Koster R. Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*. 2001. Т. 133, № 1. С. 32–43. URL: [https://doi.org/10.1016/s0377-2217\(00\)00177-6](https://doi.org/10.1016/s0377-2217(00)00177-6).

80. Hwang \* H., Oh Y. H., Lee Y. K. An evaluation of routing policies for order-picking operations in low-level picker-to-part system. *International Journal of Production Research*. 2004. Т. 42, № 18. С. 3873–3889. URL: <https://doi.org/10.1080/00207540410001696339>.

81. An exact analysis and comparison of manual picker routing heuristics / T. Engels та ін. *Queueing Systems*. 2024. URL: <https://doi.org/10.1007/s11134-024-09928-9>.
82. Using a TSP heuristic for routing order pickers in warehouses / C. Theys та ін. *European Journal of Operational Research*. 2010. Т. 200, № 3. С. 755–763. URL: <https://doi.org/10.1016/j.ejor.2009.01.036>.
83. Chan H., Pang K. Association rule based approach for improving operation efficiency in a randomized warehouse. м. Kuala Lumpur, 22–24 січ. 2011 р.
84. Çelik M., Archetti C., Süral H. Inventory routing in a warehouse: The storage replenishment routing problem. *European Journal of Operational Research*. 2021. URL: <https://doi.org/10.1016/j.ejor.2021.11.056>.
85. Surekha Ramireddy. E-commerce data set [Електронний ресурс] // Kaggle. Режим доступу: <https://www.kaggle.com/code/surekharamireddy/e-commerce-data-set/input>.
86. Aghabozorgi S., Seyed Shirkhorshidi A., Ying Wah T. Time-series clustering – A decade review. *Information Systems*. 2015. Т. 53. С. 16–38. URL: <https://doi.org/10.1016/j.is.2015.04.007>.
87. W. Kim та ін. *Data Mining and Knowledge Discovery*. 2003. Т. 7, № 1. С. 81–99. URL: <https://doi.org/10.1023/a:1021564703268>.
88. Ramanathan R. ABC inventory classification with multiple-criteria using weighted linear optimization. *Computers & Operations Research*. 2006. Т. 33, № 3. С. 695–700. URL: <https://doi.org/10.1016/j.cor.2004.07.014>.
89. Keogh E., Kasetty S. *Data Mining and Knowledge Discovery*. 2003. Т. 7, № 4. С. 349–371. URL: <https://doi.org/10.1023/a:1024988512476>.
90. Patro S. G. K., sahu K. K. Normalization: A Preprocessing Stage. *IARJSET*. 2015. С. 20–22. URL: <https://doi.org/10.17148/iarjset.2015.2305>.
91. In-Datcenter Performance Analysis of a Tensor Processing Unit / N. P. Jouppi та ін. *ISCA '17: The 44th Annual International Symposium on*

*Computer Architecture*, м. Toronto ON Canada. New York, NY, USA, 2017.  
URL: <https://doi.org/10.1145/3079856.3080246>.

92. Kohonen T. The self-organizing map. *Proceedings of the IEEE*. 1990. T. 78, № 9. C. 1464–1480. URL: <https://doi.org/10.1109/5.58325>.

93. Ward J. H. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*. 1963. T. 58, № 301. C. 236–244. URL: <https://doi.org/10.1080/01621459.1963.10500845>.

94. Petitjean F., Ketterlin A., Gançarski P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*. 2011. T. 44, № 3. C. 678–693. URL: <https://doi.org/10.1016/j.patcog.2010.09.013>.

95. Finding a "Kneedle" in a Haystack: Detecting Knee Points in System Behavior / V. Satopaa та ін. *2011 31st International Conference on Distributed Computing Systems Workshops (ICDCS Workshops)*, м. Minneapolis, MN, USA, 20–24 черв. 2011 р. URL: <https://doi.org/10.1109/icdcs.2011.20>.

96. Rousseeuw P. J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*. 1987. T. 20. C. 53–65. URL: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7).

97. Boysen N., de Koster R., Weidinger F. Warehousing in the e-commerce era: A survey. *European Journal of Operational Research*. 2019. T. 277, № 2. C. 396–411. URL: <https://doi.org/10.1016/j.ejor.2018.08.023>.

98. Diefenbach H., Grosse E. H., Glock C. H. Human-and-cost-centric storage assignment optimization in picker-to-parts warehouses. *European Journal of Operational Research*. 2024. URL: <https://doi.org/10.1016/j.ejor.2024.01.033>.

99. Redmer A. Analysis of the length of order-picking paths determined using the S-shape method. *Logforum*. 2020. T. 16, № 1. C. 33–46. URL: <https://doi.org/10.17270/j.log.2020.379>.

100. Order Batching Problems: taxonomy and literature review / E. G. Pardo та ін. *European Journal of Operational Research*. 2023. URL: <https://doi.org/10.1016/j.ejor.2023.02.019>.

101. Aboelfotoh A., Singh M., Suer G. Order Batching Optimization for Warehouses with Cluster-Picking. *Procedia Manufacturing*. 2019. T. 39. C. 1464–1473. URL: <https://doi.org/10.1016/j.promfg.2020.01.302>.

102. Parikh P. J., Meller R. D. Selecting between batch and zone order picking strategies in a distribution center. *Transportation Research Part E: Logistics and Transportation Review*. 2008. T. 44, № 5. C. 696–719. URL: <https://doi.org/10.1016/j.tre.2007.03.002>.

103. McGill R., Tukey J. W., Larsen W. A. Variations of Box Plots. *The American Statistician*. 1978. T. 32, № 1. C. 12–16. URL: <https://doi.org/10.1080/00031305.1978.10479236>.

104. Lakens D. Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. *Frontiers in Psychology*. 2013. T. 4. URL: <https://doi.org/10.3389/fpsyg.2013.00863>.

105. Post-hoc tests in one-way ANOVA: The case for normal distribution / J. Juarros-Basterretxea та ін. *Methodology*. 2024. T. 20, № 2. C. 84–99. URL: <https://doi.org/10.5964/meth.11721>.



## Додаток А – Лістинг коду алгоритму підготовки даних та виділення Pareto-групи

```

from google.colab import drive
import pandas as pd
import numpy as np
from sklearn.preprocessing import MinMaxScaler
import os

# Підключення Google Drive
drive.mount('/content/drive')
%cd /content/drive/MyDrive/postgraduate_studies/E-commerce_data_set

# Завантаження даних
df = pd.read_csv('data.csv', encoding='ISO-8859-1')

# Список службових кодів
service_codes = [
    'POST', 'BANK CHARGES', 'DOT', 'DOTCOM', 'AMAZONFEE',
    'CRUK', 'PADS', 'DCGSSBOY', 'DCGSSGIRL', 'B', 'D', 'M', 'S', 'm'
]

# Очищення даних
df_clean = df[
    (df['Quantity'] > 0) &
    df['CustomerID'].notna() &
    df['Description'].notna() &
    (~df['StockCode'].isin(service_codes))
].copy()

# Обчислення TotalSales
df_clean['TotalSales'] = df_clean['Quantity'] * df_clean['UnitPrice']

# Pareto Head
sales_by_item =
df_clean.groupby('StockCode')['TotalSales'].sum().sort_values(ascending=False)
cumulative_sales = sales_by_item.cumsum()
total_sales = sales_by_item.sum()
pareto_threshold = 0.8 * total_sales
pareto_items = cumulative_sales[cumulative_sales <= pareto_threshold].index.tolist()

print(f"Кількість товарів у Pareto Head: {len(pareto_items)}")

# Збереження Pareto Head у файл
pareto_df = sales_by_item.loc[pareto_items].reset_index()
pareto_df.columns = ['StockCode', 'Сума з TotalSales']
pareto_df.to_excel('pareto_head_items.xlsx', index=False)

# Часова трансформація
df_clean['InvoiceDate'] = pd.to_datetime(df_clean['InvoiceDate'])
df_clean['MinutesOfYear'] = (
    (df_clean['InvoiceDate'].dt.dayofyear - 1) * 24 * 60 +
    df_clean['InvoiceDate'].dt.hour * 60 +
    df_clean['InvoiceDate'].dt.minute

```

)

# Моно-вимірні датафрейми

item\_dataframes = {}

max\_length = 0

for item in pareto\_items:

    item\_df = df\_clean[df\_clean['StockCode'] == item][['Quantity',  
'MinutesOfYear']].reset\_index(drop=True)

    max\_length = max(max\_length, len(item\_df))

    item\_dataframes[item] = item\_df

# Padding

for item, df\_item in item\_dataframes.items():

    pad\_size = max\_length - len(df\_item)

    if pad\_size > 0:

        pad\_df = pd.DataFrame({'Quantity': [0]\*pad\_size, 'MinutesOfYear': [0]\*pad\_size})

        item\_dataframes[item] = pd.concat([df\_item, pad\_df], ignore\_index=True)

# Нормалізація Quantity

scaler = MinMaxScaler()

for item in item\_dataframes:

    item\_dataframes[item]['Quantity'] = scaler.fit\_transform(item\_dataframes[item][['Quantity']])

## Додаток Б – Лістинг коду формування ознак, PCA та агломеративної кластеризації товарів

```

!pip -q install openpyxl scipy matplotlib

from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.cluster import AgglomerativeClustering
from sklearn.metrics import silhouette_score
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.cluster.hierarchy import linkage, dendrogram, fcluster

# ----- НАЛАШТУВАННЯ -----
FREQ = "M"          # агрегація за місяцями
USE_LOG1P = True    # лог-трансформація TotalSales
PCA_VAR_RATIO = 0.95 # цільова частка збереженої варіації даних
N_CLUSTERS = 16     # ціль для кластеризації sklearn
DENDRO_P = 20       # скільки кластерів показувати в дендрограмі

df_ph = df_clean[df_clean['StockCode'].isin(pareto_items)].copy()
df_ph['InvoiceDate'] = pd.to_datetime(df_ph['InvoiceDate'])

# Ознаки: SKU × час (сума TotalSales за місяць)
ts = (
    df_ph.groupby(['StockCode', pd.Grouper(key='InvoiceDate', freq=FREQ)]['TotalSales']
        .sum()
        .reset_index()
)

X = (
    ts.pivot(index='StockCode', columns='InvoiceDate', values='TotalSales')
        .fillna(0.0)
        .astype(float)
)

X = X.reindex(index=pareto_items).fillna(0.0)

print(f"Матриця ознак: {X.shape[0]} SKU × {X.shape[1]} періодів")

# Трансформації, масштабування, PCA
if USE_LOG1P:
    X = np.log1p(X)

scaler = StandardScaler()
Xs = scaler.fit_transform(X)

# Безпечні межі для кількості компонент
n_comp_max = max(2, min(Xs.shape[0]-1, Xs.shape[1]))
pca = PCA(n_components=n_comp_max, svd_solver="full", random_state=42)
Xp = pca.fit_transform(Xs)

# залишаємо мінімум компонент, що дають >= 95% дисперсії (але не менше 2)

```

```

cum = np.cumsum(pca.explained_variance_ratio_)
k = int(np.searchsorted(cum, PCA_VAR_RATIO) + 1)
k = max(2, min(k, n_comp_max))
Xp = Xp[:, :k]
print(f"PCA компоненти: {k}")

# Agglomerative (Ward) з евклідовою відстанню
agg = AgglomerativeClustering(n_clusters=N_CLUSTERS, linkage="ward")
labels_sklearn_16 = agg.fit_predict(Xp)

# Silhouette (euclidean)
sil = silhouette_score(Xp, labels_sklearn_16, metric="euclidean")
print(f"Silhouette (euclidean): {sil:.6f}")

# Збереження результату (16 кластерів, sklearn)
out16 = pd.DataFrame({
    "StockCode": X.index, # ті самі SKU з Pareto Head
    "Cluster_16": labels_sklearn_16
}).reset_index(drop=True)

out16_xlsx = "clusters_16_agglomerative_ward.xlsx"
out16.to_excel(out16_xlsx, index=False)

print("Збережено (16 кластерів, sklearn):")
print("- XLSX:", out16_xlsx)

# linkage('ward') за визначенням використовує евклідову геометрію;
# це відтворює ту ж ієрархію, яку оптимізує AgglomerativeClustering(linkage='ward')
Z = linkage(Xp, method='ward')

# Горизонтальна лінія для 16 кластерів (висота злиття, що дає k=16)
k_line = 16
threshold_16 = Z[-(k_line - 1), 2]

# Розмітка для 20 кластерів
k_out = 20
labels_20_from_Z = fcluster(Z, t=k_out, criterion='maxclust')

# Додатково: розмітка для 16 кластерів з дендрограми – перевіримо розподіл
labels_16_from_Z = fcluster(Z, t=k_line, criterion='maxclust')

# Перевірка: розподіли за кластерами (розміри) мають збігатися з sklearn-версією (може
відрізнатися лише номер кластера)
sizes_sklearn = pd.Series(labels_sklearn_16).value_counts().sort_index()
sizes_from_Z = pd.Series(labels_16_from_Z).value_counts().sort_index()
print("\nПеревірка розподілу (16 кластерів):")
print("sklearn:", list(sizes_sklearn))
print("from Z :", list(sizes_from_Z))

# Дендрограма: показати 20 «листових» кластерів + лінія для k=16
plt.figure(figsize=(7, 4))
dendrogram(Z,
    truncate_mode='lastp', p=DENDRO_P, # показуємо останні 20 кластерів
    show_leaf_counts=True,
    leaf_rotation=90, leaf_font_size=9,
    color_threshold=None)

```

```

plt.axhline(y=threshold_16, color='red', linestyle='--', label='k = 16')
plt.title('Hierarchical Clustering Dendrogram')
plt.ylabel('Distance')
plt.legend()
plt.tight_layout()
plt.show()

```

# Збереження результатів

# Додамо описи та TotalSales для зручності аналізу

```

desc_map = (df_clean[df_clean['StockCode'].isin(pareto_items)]
            .drop_duplicates('StockCode')[['StockCode', 'Description']])

```

```

result20 = (pd.DataFrame({'StockCode': X.index, 'Cluster_20': labels_20_from_Z})
            .merge(desc_map, on='StockCode', how='left')
            .merge(sales_by_item.rename('TotalSales'),
                  left_on='StockCode', right_index=True, how='left')
            .sort_values('Cluster_20')
            .reset_index(drop=True))

```

```

result20_path = 'sku_clusters_20_ahc_ward.xlsx'

```

```

result20.to_excel(result20_path, index=False)

```

```

print('Збережено. Дані, згруповані в 20 кластерів:', result20_path)

```

## Додаток В – Лістинг коду побудови часових рядів і кластеризації з визначенням оптимального числа кластерів

```
# Побудова матриці часових рядів і кластеризація
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
import numpy as np

# Формуємо матрицю X (рядки – SKU, стовпці – Quantity у часі)
X = np.vstack([df_item['Quantity'].to_numpy() for df_item in item_dataframes.values()])

# Нормалізація по рядках (z-score), щоб кластеризація була за формою
X_scaled = (X - X.mean(axis=1, keepdims=True)) / (X.std(axis=1, keepdims=True) + 1e-8)

# Elbow Method + Delgado rule
MAX_K = min(60, X_scaled.shape[0] - 1) # щоб k не перевищував кількість SKU
Ks = list(range(2, MAX_K + 1))
wcss = []

for k in Ks:
    kmeans = KMeans(n_clusters=k, n_init='auto', random_state=42)
    kmeans.fit(X_scaled)
    wcss.append(kmeans.inertia_)

# Delgado rule: найбільша відстань до прямої між крайніми точками
pts = np.column_stack((Ks, wcss))
p1, p2 = pts[0], pts[-1]
line_vec = p2 - p1

def dist_to_line(point, start=p1, line=line_vec):
    vec = point - start
    return abs(np.cross(line, vec)) / np.linalg.norm(line)

distances = [dist_to_line(pt) for pt in pts]
elbow_k = int(pts[np.argmax(distances)][0])

print(f"Оптимальна кількість кластерів (метод Delgado): k = {elbow_k}")

# Візуалізація з помаранчевою штриховою лінією між крайніми точками
# Пряма між крайніми точками
y_line = np.linspace(wcss[0], wcss[-1], num=len(Ks))

plt.figure(figsize=(7, 4))
plt.plot(Ks, wcss, 'o-', label='WCSS')
plt.plot(Ks, y_line, '--', color='orange', label='Пряма між крайніми точками')
plt.axvline(elbow_k, color='red', linestyle='--', label=f'Elbow k={elbow_k}')
plt.title('Elbow method (WCSS vs k)')
plt.xlabel('Кількість кластерів k')
plt.ylabel('WCSS')
plt.grid(True, alpha=0.3)
plt.legend()
plt.show()
```

## Додаток Г – Лістинг коду ANOVA для групових порівнянь і пост-хок аналізу Tukey HSD

```
!pip -q install pandas numpy scipy statsmodels openpyxl

import io, numpy as np, pandas as pd
from scipy import stats
from statsmodels.stats.multicomp import pairwise_tukeyhsd
from google.colab import files

ALPHA = 0.05
SHEET = "data"
GROUP_COL = "policy"
VALUE_COL = "distance_m"

# Завантаження Excel в сесію
uploaded = files.upload()
fname = next(iter(uploaded))
assert fname.lower().endswith((".xlsx", ".xls")), "Очікую Excel-файл (.xlsx/.xls)"

# Читання потрібного аркуша і сувора перевірка колонок
df = pd.read_excel(io.BytesIO(uploaded[fname]), sheet_name=SHEET)
required = {GROUP_COL, VALUE_COL}
if not required.issubset(df.columns):
    raise ValueError(f"У файлі мусять бути колонки: {sorted(required)}")

# Підготовка даних
df = (df[[GROUP_COL, VALUE_COL]]
      .dropna()
      .assign(**{GROUP_COL: lambda x: x[GROUP_COL].astype(str)}))
df[VALUE_COL] = pd.to_numeric(df[VALUE_COL], errors="coerce")
df = df.dropna()

# Описова статистика (опційно для контролю)
print("Розміри груп і середні значення:")
print(df.groupby(GROUP_COL)[VALUE_COL].agg(["count", "mean", "std"]).round(3), "\n")

# One-way ANOVA
groups = [g[VALUE_COL].values for _, g in df.groupby(GROUP_COL)]
F, p = stats.f_oneway(*groups)
anova_df = pd.DataFrame({"F-stat": [F], "p-value": [p]})
print("ANOVA")
print(anova_df.round(6), "\n")

# Якщо ANOVA значуща – попарні порівняння Tukey HSD
if p < ALPHA:
    res = pairwise_tukeyhsd(endog=df[VALUE_COL].values,
                           groups=df[GROUP_COL].values,
                           alpha=ALPHA)
    tukey_df = pd.DataFrame(res._results_table.data[1:], columns=res._results_table.data[0])
    tukey_df = tukey_df.astype({"meandiff": "float64", "p-adj": "float64", "lower": "float64", "upper": "float64", "reject": "bool"})
    print("Tukey")
    print(tukey_df.round(4))
else:
    print("Tukey HSD не виконуємо (ANOVA незначуща).")
```