



В. Р. Турчак¹, О. В. Овсяк²

¹ Національний університет "Львівська політехніка", м. Львів, Україна

² Національний лісотехнічний університет України, м. Львів, Україна

МОДЕЛЬ АЛГОРИТМУ БАГАТОКРИТЕРІАЛЬНОГО СОРТУВАННЯ ДАНИХ З ЇХ ДИНАМІЧНО-НАСЛІДКОВИМ ОБМЕЖЕННЯМ ПЕРЕМІЩЕНЬ

Багатокритеріальне сортування структурованих даних із збереженням часткового порядку сортування та стабільності їх впорядкування є ключовим завданням у сучасних аналітичних системах, що застосовують в електронній комерції, рекомендаційних сервісах, медичних платформах, системах управління персоналом, логістиці, освітніх платформах, фінансових аналітичних системах й обробленні великих обсягів структурованих даних для прийняття рішень. У дослідженні розроблено інноваційну модель алгоритму динамічно-наслідкового сортування даних, що забезпечує гнучкість ранжування даних різних типів. Алгоритм базується на модифікованій версії Selection Sort, він гарантує стабільність впорядкування структурованих даних за основними критеріями навіть за впливу другорядних факторів, завдяки чіткому визначенню меж динамічного переміщення структурованих даних з урахуванням ієрархії критеріїв у процесі ранжування даних. Досліджено вплив параметра переміщення структурованих даних на збереження часткового порядку та обмеження переміщень об'єктів у масивах даних різних типів, враховуючи числові і гібридні їхні набори. Встановлено, що створений алгоритм забезпечує ефективне зменшення переміщень структурованих даних, зберігаючи стабільність їх ранжування в умовах динамічних змін пріоритетів критеріїв. Виявлено кореляцію між кількістю критеріїв і рівнем стабільності, що сприяє оптимізації процедури сортування даних для різних прикладних завдань. Підтверджено переваги розробленого алгоритму над алгоритмами Multi-Key Sort, Weighted Score Sort і ELECTRE за адаптивністю, інтерпретованістю і стабільністю. Гнучке налаштування пріоритетів критеріїв робить можливим адаптувати алгоритм до специфічних вимог таких завдань, як ранжування даних товарів, оцінка медичних даних, оптимізація логістичних процесів чи аналіз освітніх показників. Розроблена модель алгоритму є релевантною для аналітичних систем, які потребують адаптивного, контрольованого та інтерпретованого ранжування даних, і відкриває перспективи для подальшої оптимізації в завданнях оброблення великих обсягів даних, інтеграції в системи реального часу, масштабування для високонавантажених платформ, а також адаптації до унікальних вимог прикладних доменів, таких як персоналізовані рекомендації, автоматизоване прийняття рішень чи аналіз фінансових ризиків.

Ключові слова: частковий порядок; стабільність ранжування даних; пріоритетність критеріїв; адаптивне сортування структурованих даних; стабільність впорядкування.

Вступ / Introduction

Багатокритеріальне сортування структурованих даних є ключовим компонентом систем багатокритеріального прийняття рішень MCDM (англ. *Multi-Criteria Decision-Making*), зокрема у складі гібридних систем підтримки прийняття рішень. Традиційні алгоритми, такі як QuickSort і MergeSort, непридатні для багатокритеріальних задач через повну перебудову масиву структурованих даних. Сучасні алгоритми, зокрема Multi-Key Sort, Weighted Score Sort, ELECTRE, TOPSIS і AHP, забезпечують гнучкість ранжування даних, але мають істотний недолік: нестабільність, що проявляється в надмірних переміщеннях об'єктів за зміни пріоритетів чи параметрів. Алгоритм Multi-Key Sort втрачає стабільність через послідовне сортування структурованих даних, алгоритм Weighted Score Sort чутливий до вагових коефіцієнтів (наприклад, 0,5 для ціни; 0,3 для якості),

які визначають важливість критеріїв, і їх зміна може переставити об'єкти в сортуванні, бо оцінки множаться на ці ваги, тож неточні ваги спотворюють результати. Алгоритми ELECTRE та TOPSIS мають складність у налаштуванні порогових значень вагових коефіцієнтів, що визначають, як порівнюються об'єкти. Наприклад, в ELECTRE: якщо ціна першого товару менша за ціну другого на 10 грн – перший вважають кращим, а якщо різниця становить тільки 2 грн – товари майже еквівалентні. У TOPSIS: товар, дорожчий на 5 % від найкращого, все ще можна вважати прийнятним. Проте ці алгоритми не забезпечують контроль за локальними переміщеннями структурованих даних, оскільки незначні зміни в числових значеннях або невдало обрані пороги можуть спотворити результати, роблячи їх нестабільними та ненадійними, а алгоритм AHP – обчислювально складним. Наприклад, адаптація алгоритму ELECTRE

Інформація про авторів:

Турчак Володимир Романович, аспірант, кафедра комп'ютерних технологій у видавничо-поліграфічних процесах.

Email: volodymyr.turchak@gmail.com; <https://orcid.org/0009-0009-2907-1869>

Овсяк Олександр Володимирович, д-р техн. наук, професор, кафедра комп'ютерних наук.

Email: ovsjak@ukr.net; <https://orcid.org/0000-0003-2620-1938>

Цитування за ДСТУ: Турчак В. Р., Овсяк О. В. Модель алгоритму багатокритеріального сортування даних з їх динамічно-наслідковим обмеженням переміщень. Науковий вісник НЛТУ України. 2025, т. 35, № 3. С. 175–182.

Citation APA: Turchak, V. R., & Ovsyak, O. V. (2025). Model of a multicriteria data sorting algorithm with dynamic-consequential movement constraints. *Scientific Bulletin of UNFU*, 35(3), 175–182. <https://doi.org/10.36930/40350319>

для оцінювання ефективності сортування даних не усуває проблему нестабільності результатів – незначні зміни у вхідних даних або порогах можуть істотно змінити висновки. Так, зміна оцінки за критерієм "вартість" на 1 % може перемістити проєкт із категорії "хороший" до "середній". Хоча ELECTRE застосовують для ранжування даних варіантів за критеріями (вартість, терміни, ризики тощо), він чутливий до налаштувань порогів переваги та байдужості, що робить його ненадійним для сортування даних у динамічних умовах. Бібліографічний аналіз наукових публікацій підтверджує зростання інтересу до гібридних MCDM, які поєднують стабільність й адаптивність. Сучасні роботи свідчать про потребу в алгоритмах із локальним коригуванням і контролем переміщень структурованих даних для підвищення їхньої інтерпретованості та довіри користувачів.

Багатокритеріальне сортування структурованих даних є ключовим елементом сучасних інформаційних технологій і систем, що забезпечують опрацювання даних у сферах електронної комерції, рекомендаційних сервісів, медичних платформ, логістики, управління персоналом, освіти та фінансової аналітики, це зумовлює актуальність дослідження. Зростання обсягів даних і потреба в персоналізованих рішеннях вказують на важливість алгоритмів, які поєднують адаптивність, стабільність і контроль переміщень структурованих даних. Такі моделі застосовують для оцінювання проєктів, аналізу освітніх даних і медичних реєстрів, що підтверджує їхню релевантність.

Об'єкт дослідження – багатокритеріальне сортування масивів структурованих даних.

Предмет дослідження – методи і засоби багатокритеріального сортування даних з динамічно-наслідковим обмеженням їх переміщення, що дасть змогу зменшити надмірні їх переміщення у масиві відсортованих даних і гарантує стабільність отриманих результатів сортування.

Мета роботи – розробити алгоритм адаптивного багатокритеріального сортування даних, який забезпечить збереження часткового їх порядку у масиві відсортованих даних і стабільність отриманих результатів сортування шляхом введення динамічно-наслідкового обмеження на їх переміщення.

Для досягнення зазначеної мети визначено такі основні завдання дослідження:

- проаналізувати відомі моделі алгоритмів багатокритеріального сортування структурованих даних у контексті стабільності і динамічного обмеження переміщень даних, що дасть змогу визначити їхні переваги та недоліки, сформулювати обмеження традиційних підходів, окреслити вимоги до алгоритмів, здатних зберігати частковий порядок та стабільність отриманих результатів сортування;
- розробити алгоритм адаптивного багатокритеріального сортування структурованих даних з динамічно-наслідковим обмеженням переміщенням даних, який враховує пріоритетність критеріїв й описує процедуру сортування структурованих даних у режимах зростання, спадання та ідентифікації збігу значень параметрів даних, що забезпечить збереження часткового порядку структурованих даних, зменшить надмірні їх переміщення у масиві відсортованих даних і гарантує стабільність отриманих результатів сортування;
- провести порівняння ефективності роботи розробленого алгоритму з відомими алгоритмами сортування структурованих, що дасть змогу оцінити його переваги за збереження часткового порядку та мінімізації обсягу переми-

щень, а також продемонструвати його ефективність застосування у прикладних алгоритмічних задачах.

Аналіз останніх досліджень та публікацій. Під час аналізу відомих алгоритмів багатокритеріального сортування структурованих даних встановлено, що в сучасних інформаційних системах і технологіях – аналітика, електронна комерція, рекомендаційні платформи – зростає потреба у сортуванні за числовими (ціна, рейтинг) та категоріальними (наявність, тип) критеріями з акцентом на стабільність отриманих результатів і контроль локальних переміщень. Останні дослідження в галузі багатокритеріального прийняття рішень MCDM (англ. *Multiple Criteria Decision Making*) вказують на істотні обмеження наявних методів у частині забезпечення стабільності їх ранжування, адаптивності до змін вагових коефіцієнтів та збереження часткового порядку. Традиційні алгоритми, зокрема QuickSort і MergeSort, не враховують цих особливостей, є чутливими до незначних змін у вхідних даних і не забезпечують інтерпретованих результатів.

У роботі [2] розглянуто сучасні методи MCDM-сортування даних, зокрема Multi-Key Sort та Weighted Score Sort. Перший забезпечує поетапне сортування за критеріями, але демонструє низьку стабільність за зміни їх порядку – переміщення можуть охоплювати до 60 % об'єктів. Weighted Score Sort чутливий до змін вагових коефіцієнтів: варіація на 10 % здатна істотно змінити результат. Запропонований алгоритм демонструє стабільність $\tau = 0,92 - 0,95$ та зменшує середнє переміщення до 2,1-2,5 позицій, що вигідно вирізняє його серед зазначених підходів.

У дослідженні [3] аналізують методи ELECTRE, TOPSIS і AHP. ELECTRE потребує ретельного налаштування порогів переваги, TOPSIS ґрунтується на евклідових відстанях, а AHP має квадратичну обчислювальну складність $O(n^2)$, що обмежує його застосування у задачах великої розмірності. Натомість запропонований алгоритм, побудований на модифікованому Selection Sort, забезпечує локальну стабільність без складного параметричного налаштування.

У бібліографічному аналізі інформації вказують на зростання зацікавленості до гібридних підходів, які поєднують сортування даних, їх класифікацію та кластеризацію. Водночас автори зазначають, що наявні методи сортування даних рідко забезпечують контроль локальних переміщень у динамічних умовах [4]. Запропонована модель алгоритму враховує цю потребу, скорочуючи середню амплітуду переміщень на 50–60 % порівняно з класичними алгоритмами.

У роботі [9] запропоновано метод Best-Worst PROMETHEE для оцінювання продуктивності шкіл у рамках проєкту PISA від OECD, з акцентом на багатокритеріальне сортування. Автори зазначають, що традиційний PROMETHEE є чутливим до суб'єктивного вибору вагових коефіцієнтів, що може призводити до нестабільності їх ранжування, і пропонують інтеграцію з Best-Worst Method для зменшення цієї проблеми. Розроблений алгоритм усуває подібні обмеження, забезпечуючи стабільність і контроль переміщень.

У роботі [15] зацентовано увагу на потребі сортувальних алгоритмів, здатних адаптуватись до змін пріоритетів у сферах логістики та електронної торгівлі. Автори зазначають, що більшість наявних рішень не передбачають регулювання інтенсивності змін у ре-

зультатах ранжування даних. Запропонована модель алгоритму із параметром R вирішує цю проблему, забезпечуючи адаптивне та стабільне сортування даних, підтверджене експериментально $\tau = 0,92 - 0,95$.

Метод PROMETHEE, розглянутий у роботі [6], реалізує некомпенсаційний підхід і забезпечує ранжування даних на підставі ординальних входів. Проте модель є чутливою до змін вагових коефіцієнтів і не передбачає механізму обмеження локальних переміщень. Розроблений алгоритм, навпаки, гарантує інтерпретовану стабільність навіть за зміни другорядних факторів, завдяки чіткому регулюванню зони допустимих переміщень.

Проведений аналіз останніх досліджень і публікацій стосовно наявних MCDM-методів свідчить про практичну їхню відсутність, які одночасно забезпечують стабільність ранжування даних, збереження часткового порядку та контроль локальних переміщень структурованих даних. Більшість відомих реалізацій орієнтовані на фіксовані сценарії, наприклад, оцінювання альтернатив з наперед заданими критеріями, або повністю перебудовують масив за зміни вагових коефіцієнтів, що ускладнює застосування в динамічних середовищах, де критичними є стабільність отриманих результатів і швидкість реагування. Виявлена розбіжність між вимогами прикладних завдань і можливостями наявних методів обґрунтовує потребу у створенні методу адаптивного багатокритеріального сортування з динамічно-наслідковим обмеженням переміщень, який забезпечує збереження часткового порядку та контроль змін у впорядкуванні без повної перебудови масиву структурованих даних.

Матеріали та методи дослідження. Дослідження виконано над структурованими даними, які містять різ-

нотипні параметри (чутливість, дальність, рейтинг, ціна, колір та інші). Зокрема, набір e-commerce ($N = 100$, $k = 3$: ціна, рейтинг, тип товару) зібрано з платформи онлайн-продажів [7]. Дані HR-відбору ($N = 120$, $k = 5$: оцінювання продуктивності, спеціалізація, освіта, досвід, вік) отримано з набору IBM HR Analytics [17]. Медичні записи ($N = 150$, $k = 4$: числові показники, рівень ризику) отримано з анонімованих баз MIMIC-III [10].

Теорію алгоритмів використано для створення моделі алгоритму з динамічно-наслідковим обмеженням переміщенням структурованих даних [5]. Експерименти виконано в Python 3,8 у середовищі Jupyter Notebook. Використано бібліотеки NumPy (1.19.5) і Pandas (1.2.4) для оброблення даних, Matplotlib (3.3.4) – для створення графіків [8] і модуль time – для вимірювання тривалості виконання. У методі порівняння застосовано реальні дані. Для оцінювання продуктивності застосовано коефіцієнт Кендалла, середнє переміщення структурованих даних і тривалість виконання багатокритеріального сортування структурованих даних.

Результати дослідження та їх обговорення / Research results and their discussion

Розроблено алгоритм адаптивного багатокритеріального сортування структурованих даних з динамічно-наслідковим обмеженням переміщення даних, який враховує пріоритетність критеріїв й описує процедуру сортування структурованих даних у режимах зростання, спадання та ідентифікації збігу значень параметрів даних, що забезпечує збереження часткового порядку структурованих даних, зменшує надмірні їх переміщення у масиві відсортованих даних і гарантує стабільність отриманих результатів сортування (рис. 1).

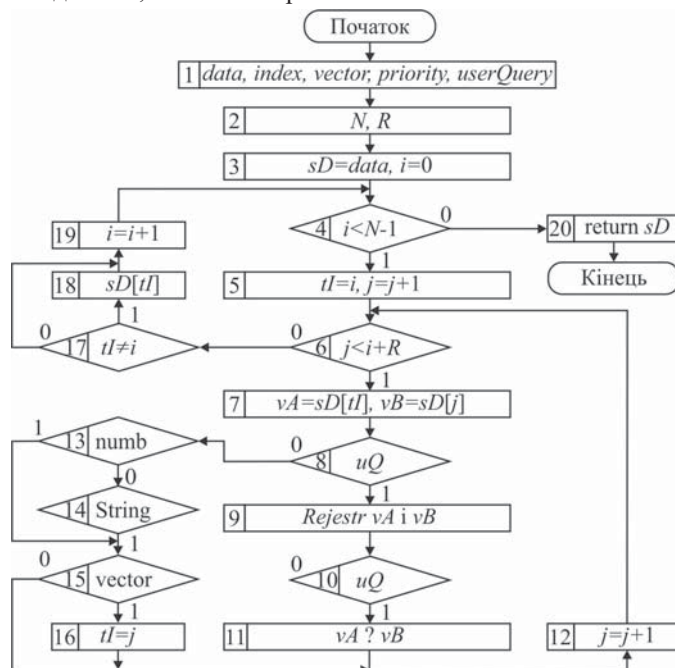


Рис. 1. Блок-схема алгоритму адаптивного багатокритеріального сортування структурованих даних з динамічно-наслідковим обмеженням їх переміщенням / Flowchart of the adaptive multicriteria sorting algorithm for structured data with dynamic-consequential movement restriction

Блок 1 містить задання початкових даних, якими є:

- $data$ (d) – двовимірний масив вхідних структурованих даних, який утворений рядками з декількома заданими даними (ними, наприклад, може бути фірма виробник товару, технічні характеристики виробу, ціна тощо), а структурованими даними рядків утворені колонки;

- $index$ (i) – змінна рядків масиву $data$ (d);
- $vector$ (v) – тип сортування структурованих даних ("asc" – сортування структурованих даних за зростанням (збільшенням значення), "desc" – сортування структурованих даних за спаданням (зменшенням значень), "max"/"min" – пошук максимального/мінімального значень, "match" – пошук конкретного збігу);

- *priority (p)* – пріоритет (або перелік пріоритетів, який містить z впорядкованих натуральних чисел). Позначають як p_1, p_2, \dots . Найменша кількість пріоритету означає найвищу важливість (наприклад, p_1 – найвищий). Перелік пріоритетів задає порядок важливості критеріїв, якщо критеріїв декілька, алгоритм послідовно "проходить" пріоритетами від найвищого до найнижчого;
- *userQuery (uQ)* – текстовий запит (для сортування структурованих даних типу "match").

Автоматичне встановлення кількості рядків (N), яку містить вхідний масив структурованих даних (елементів), здійснюється на другому етапі виконання алгоритму (блок 2). Значення N використовують під час налаштування лімітів та індексації. Також у блоці 2 обчислюється значення обмеження (R) на переміщення рядка на максимальну кількість позицій, на яку може переміститися рядок у процесі ранжування даних за пріоритетом.

Щоб зберегти вхідні дані без їхніх змін створюють копію (sD) масиву d (*data*) і задають початкове значення змінної ($i = 0$), що описано у блоці 3 алгоритму. Сортування даних розпочинається з найвищого пріоритету.

У блоці 4 запускають зовнішній цикл за індексом i . Цикл проходить через усі позиції в масиві від 0 до $N - 1$. Завдання циклу: знайти та розмістити "кращий" елемент для кожної позиції i . У блоці 4 виконується перевірка на допустиму задану кількість рядків ($i \leq (N - 1)$). Якщо вона не перевершена, то у блоці 5 вводиться змінна *targetIdx (tI)*, якій приписується значення змінної (i), а змінній рядків обмеження переміщення (j) встановлюється значення $i + 1$. Тим самим встановлюється початковий "найкращий" елемент, який знаходиться за змінною tI .

У блоці 6 запускається внутрішній цикл за j , починаючи з $i + 1$. Перевіряється наступні R елементів, щоб знайти кращий елемент для позиції i . Завдання внутрішнього циклу порівняти поточний "найкращий" елемент з наступними елементами в межах обмеження на переміщення R .

Якщо поточне значення змінної j не більше кількості рядків вхідного масиву ($N - 1$) або кількості рядків обмеження переміщення рядків (R), що описано у блоці 6, то виконується перехід до виконання 7 блоку. У ньому змінній *valA (vA)* приписується значення даних, яке знаходиться у вхідному масиві під адресою tI , а змінній *valB (vB)* приписується значення даних, яке знаходиться у вхідному масиві під адресою j . Отримуємо значення для порівняння значень vA і vB .

Далі, у блоці 8, описана перевірка умов чи вибраний режим сортування "match" і чи наданий текстовий запит (uQ) не порожній. Якщо це так, то у блоці 9 весь текст перетворюється до малих літер (нижній регістр). Потім, у блоці 10, встановлюється чи містить змінна vB текстовий запит і при цьому vA його не містить. Якщо це так, то змінній tI задається значення змінної j (блок 11). Після цього, як і в разі невиконання умови блоку 10, здійснюється перехід до збільшення значення змінної j (блок 12) і повернення у цикл за змінною j .

Якщо ж умова блоку 8 не виконується і значення змінних vA і vB є числовими (блок 13), то у блоці 14 виконується порівняння числових значень (*max/min*). Якщо сортування структурованих даних типу "max", то шукаємо більше кількість ($vB > vA$). Якщо ж сортування структурованих даних типу "min", то шукаємо менше кількість ($vB < vA$). Якщо умова блоку 14 істинна, то в

блоці 15 оновлюємо tI ($tI = j$), тобто знаходимо новий "найкращий" елемент. Після цього виконується перехід до блоку 12 і наступного кроку внутрішнього циклу.

Коли умова блоку 13 не виконується і значення змінних vA і vB є рядками (блок 16), то у блоці 17 описується порівняння рядкових значень (*asc/desc*) і для "asc" перевіряємо, чи рядок vB має стояти перед vA за алфавітом (тобто vB менше), а для "desc" – навпаки, чи рядок vB стоїть після vA (тобто vB більше). Після цього змінній tI задається значення змінної j (блок 18). Далі, як і в разі невиконання умови блоку 17, здійснюється перехід до збільшення значення змінної j (блок 12) і повернення у цикл за змінною j .

У блоці 12 збільшуємо j на одиницю. Якщо ще є елементи в межах переміщення ($j \leq i + R$), блок 6, то повертаємось до виконання 7 блоку. Якщо ні – переходимо до блоку 19. У ньому виконується перевірка та перестановка об'єктів (*swap*). Перевіряємо, чи знайшовся новий кращий елемент для позиції i (чи змінився tI за цикл). Якщо змінився ($tI \neq i$), міняємо місцями елементи на позиціях i та tI у масиві (блок 20). Якщо ні – нічого не робимо. У блоці 21 відбувається завершення зовнішнього циклу переходом до наступного i ($i = i + 1$) і перевіряється, чи пройшли всі рядки масиву ($i < (N - 1)$), блок 4. Якщо ще є рядки масиву – повернення до блоку 5. Якщо ні – цикл завершено, перехід до наступного кроку.

Отримуємо масив sD , в якому елементи відсортовано відповідно до заданих умов, але з обмеженням руху кожного елемента (максимум на відстань переміщення R), блок 22. На цьому робота алгоритму завершена.

Приклад застосування моделі алгоритму. Нижче наведено приклад масиву із шести рядків – елементи ($A - F$), що ранжуються за трьома критеріями: p_1 (рейтинг, "asc"), p_2 (колір, "match"), p_3 (ціна, "desc"). Масив даних наведено у вигляді табл. 1. Ця таблиця містить 6 рядків структурованих даних (елементів $A-F$) із критеріями: рейтинг (числа 1–6), колір (білий/чорний), ціна (числа 1–6).

Табл. 1. Початкові дані сортування / Initial sorting dataset

№	Елемент	p_1 – рейтинг, режим asc	p_2 – колір, режим match, β	p_3 – ціна, режим desc
1	A	5	білий, 0	3
2	B	2	блакитний, 6	4
3	C	3	сірий, 4	2
4	D	4	бежевий, 5	5
5	E	1	світлий, 5	1
6	F	6	чорний, 5	6

На першому кроці сортування структурованих даних відбувається за p_1 (рейтинг), режим сортування asc (тобто чим менше, тим вище в списку), $R = 6$. Оскільки $R = 6$ (тобто елементи можна переставляти будь-де), сортування даних працює як звичайне. Результат сортування структурованих даних на кроці 1 наведено у табл. 2.

Табл. 2. Результат сортування структурованих даних за критерієм p_1 / Result of sorting structured data by criterion p_1 (rating)

№	Елемент	p_1 – рейтинг, режим asc	p_2 – колір, режим match, β	p_3 – ціна, режим desc
1	E	1	світлий, 5	1
2	B	2	блакитний, 6	4
3	C	3	сірий, 4	2
4	D	4	бежевий, 5	5
5	A	5	білий, 0	3
6	F	6	чорний, 5	6

Результатом сортування структурованих даних є таке розміщення елементів: E на 1-му місці, B на 2-му, C на 3-му, D на 4-му, A на 5-му, F на 6-му.

У наступному сортуванні за p_2 (відповідність), *match*, $R = 3$. Сортування структурованих даних відбувається за відповідністю текстового значення до шаблону "білий". Чим менша текстова відстань за Левенштейном (β), тим краще. Дозволено перестановки в межах трьох позицій. A (єдиний білий) піднімається на три позиції до другого місця. Результат сортування структурованих даних за критерієм p_2 наведено у табл. 3.

Табл. 3. Результат сортування структурованих даних за критерієм p_2 / Result of sorting structured data by criterion p_2 (color)

№	Елемент	p_1 – рейтинг, режим <i>asc</i>	p_2 – колір, режим <i>match</i> , β	p_3 – ціна, режим <i>desc</i>
1	E	1	світлий, 5	1
2	A	5	білий, 0	3
3	B	2	блакитний, 6	4
4	C	3	сірий, 4	2
5	D	4	бежевий, 5	5
6	F	6	чорний, 5	6

Під час виконання сортування структурованих даних за критерієм p_3 (ціна), режим *desc* (тобто чим більша ціна, тим вища позиція в межах $R = 2$). У межах кожної підгрупи (по $R = 2$ позиції) коригується порядок за спаданням ціни, але зберігається попередній порядок настільки, наскільки дає змогу R . Об'єкт F (6) піднімається на дві позиції до 4-го місця, а D має ціну 5 і може піднятися до 3-го місця. C (2) і B (4), але не може перемістити A і E, бо $R = 2$. Залишається на місці. Дозволено перестановки в межах двох позицій. Результат сортування структурованих даних за критерієм p_3 подано у табл. 4.

Табл. 4. Результат сортування структурованих даних за критерієм p_3 / Result of sorting structured data by criterion p_3 (price)

№	Елемент	p_1 – рейтинг, режим <i>asc</i>	p_2 – колір, режим <i>match</i> , β	p_3 – ціна, режим <i>desc</i>
1	E	1	світлий, 5	1
2	A	5	білий, 0	3
3	D	4	бежевий, 5	5
4	F	6	чорний, 5	6
5	B	2	блакитний, 6	4
6	C	3	сірий, 4	2

Оскільки алгоритм з динамічно-наслідковим переміщенням сортованих даних впорядковує елементи за трьома критеріями, кожен об'єкт змінив позицію від табл. 1 до табл. 4. Об'єкт E (рейтинг 1) перемістився з 5-тої на 1-шу позицію через найвищий рейтинг і залишився там, бо колір і ціна не вплинули на його позицію. Елемент A спустився з 1-шої на 5-ту через високий рейтинг (5), піднявся на 2-гу за кольором (білий), але залишився на 2-й за ціною (3), бо ($R = 2$). Елемент D залишився на 4-й позиції за рейтингом, зсунувся на 5-ту через колір, але піднявся на 3-тю за ціною (5). Елемент F залишився на 6-й до третього сортування структурованих даних, коли піднявся на 4-ту за ціною (6). Елемент B тримався на 2-й, але зсунувся на 3-тю через колір і на 5-ту через ціну (4). Елемент C зсунувся з 3-тєї на 4-ту через колір і на 6-ту через ціну (2). Як видно із наведеного прикладу створена модель сортування структурованих даних дає змогу: покращити стабільність сортування структурованих даних, уникати радикальних змін через другорядні критерії, під час вико-

нання сортування даних зменшити кількість переміщень елементів.

Типовим представником відомих моделей (Multi-Key Sort, Weighted Score Sort, ELECTRE, TOPSIS, AHP) є алгоритм Weighted Score Sort (WSS). Тому проведемо порівняння створеного та WSS-алгоритмів [2].

Експерименти виконано в Python 3.8 у середовищі Jupyter Notebook. Використано бібліотеки NumPy (1.19.5) і Pandas (1.2.4) для оброблення даних, Matplotlib (3.3.4) – для створення графіків, модуль time – для вимірювання тривалості виконання.

Для оцінювання ефективності роботи алгоритмів використовуємо різноманітні набори даних. Зокрема, набір e-commerce ($N = 100$, $k = 3$: ціна, рейтинг, тип товару) зібрано з платформи онлайн-продажів, медичні записи ($N = 150$, $k = 4$: числові показники, рівень ризику) – з анонімованих баз, а дані HR-відбору ($N = 120$, $k = 5$: оцінювання продуктивності, спеціалізація, освіта, досвід, вік) – із набору IBM HR Analytics. Ці набори забезпечують комплексне тестування. Оскільки стабільність і контроль переміщень є ключовими для MCDM [4], використано три метрики: коефіцієнт Кендалла (τ) для вимірювання кореляції порядку ранжування даних, середнє переміщення об'єктів для оцінювання локальних змін і тривалість виконання (середнє за 10 запусків) для аналізу обчислювальної ефективності. Якщо τ наближається до 1, то ранжування даних стабільне, а низьке переміщення свідчить про керованість. Результати порівняння наведено у табл. 5.

Табл. 5. Результати порівняння створеного алгоритму і моделі WSS за критеріями стабільності, переміщення та тривалості виконання / Results of comparison between the proposed algorithm and WSS by stability, displacement, and runtime

Набір даних	Кендалла, τ	Переміщення	Тривалість, с
E-commerce ($N = 100$, $k = 3$)	0,95	2,1	$0,03^{\pm 0,01}$
	0,65	5,8	$0,01^{\pm 0,005}$
Медичні записи ($N = 150$, $k = 4$)	0,93	2,5	$0,04^{\pm 0,01}$
	0,62	5,6	$0,015^{\pm 0,005}$
HR-відбір ($N = 120$, $k = 5$)	0,92	2,3	$0,035^{\pm 0,01}$
	0,6	5,7	$0,012^{\pm 0,005}$

Згідно з даними табл. 5, розроблений алгоритм, завдяки ітеративному параметру $R = N/(i + 1)$, обмежує переміщення елементів, що забезпечує високу стабільність й адаптивність [11]. Натомість модель WSS (англ. *Weighted Score Sort*), який базується на зваженій сумі оцінок із рівномірними вагами $1/k$, є чутливим до змін пріоритетів, що призводить до значних переміщень (50 – 60 % елементів переміщуються). Якщо створена модель оптимізує частковий порядок, то Weighted Score Sort часто втрачає стабільність.

Наприклад, для набору e-commerce ($N = 100$, $k = 3$) створена модель досягнула $\tau = 0,95$, середнього переміщення 2,1 позиції та тривалості виконання $0,03^{\pm 0,01}$ с, тоді як Weighted Score Sort показав значно нижчий $\tau = 0,65$, переміщення 5,8 і час $0,01^{\pm 0,005}$ с. Якщо для медичних даних ($N = 150$, $k = 4$) створений алгоритм зберіг високий $\tau = 0,93$ і переміщення 2,5, то Weighted Score Sort мав $\tau = 0,62$ і переміщення 5,6. Аналогічно, у HR-відборі ($N = 120$, $k = 5$) розроблена модель стабільно перевершує Weighted Score Sort за τ і переміщенням. Отже, створений алгоритм демонструє кращу керованість [15].

Візуалізація результатів. На рис. 2. показано порівняння стабільності (коефіцієнт Кендалла, τ) алгоритмів динамічно-наслідкового обмеження переміщення (си-

ні круги) та Weighted Score Sort (червоні хрестики) на наборах даних E-commerce, медичні записи і HR-відбір.

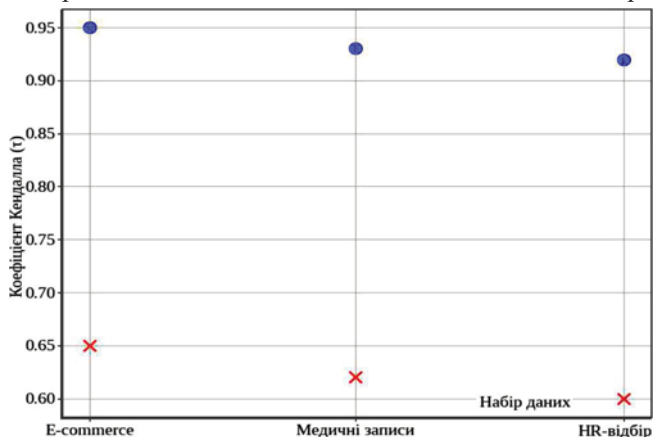


Рис. 2. Графік розсіювання коефіцієнта Кендалла, τ / Scatter plot of the Kendall τ coefficient

На осі ординат (Y) відкладено коефіцієнт Кендалла (τ), що характеризує стабільність упорядкування; величина є безрозмірною і лежить у діапазоні від 0 до 1, де вищі значення означають вищу кореляцію між очікуваним та отриманим порядком. Він показує стабільність створеної моделі ($\tau = 0,92 - 0,95$) проти Weighted Score Sort ($\tau = 0,60 - 0,65$) для всіх наборів даних та ілюструє її вищу кореляцію впорядкованості.

На рис. 3 наведено гістограму переміщень. На цьому рисунку показано розподіл середнього переміщення елементів створеного алгоритму (блакитні стовпці) і алгоритму Weighted Score Sort (червоні стовпці) в. Вісь абсцис (X) відображає середнє переміщення елементів (у позиціях) – кількість позицій, на яку елементи переміщуються від їхнього початкового положення.

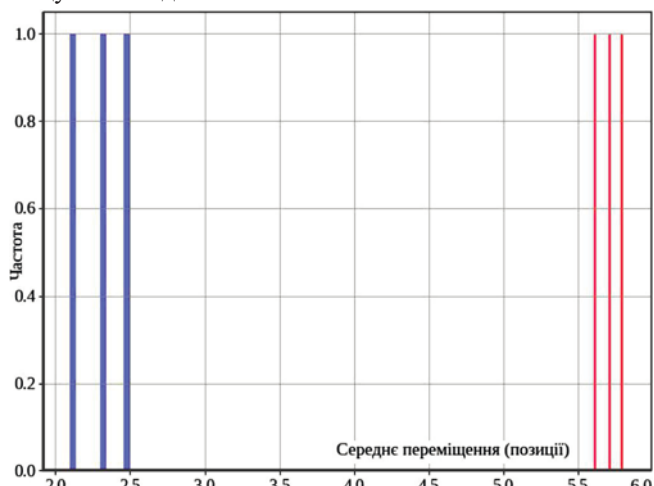


Рис. 3. Гістограма переміщень елементів / Histogram of displacements

Вісь ординат (Y) показує частоту появи відповідного середнього переміщення у вибірці (кількість наборів даних із таким значенням). Вона демонструє, що створена модель обмежує переміщення до 2,1-2,5 позицій, тоді як Weighted Score Sort має пік на 5,5-5,8. Цим самим розроблена модель скорочує переміщення елементів на 50-60 %.

Стовпцевий графік (рис. 4) тривалості виконання сортування структурованих даних ілюструє: на осі абсцис (X) подано назви наборів даних – E-commerce, Медичні записи і HR-відбір. На осі ординат (Y) відкладено середню тривалість виконання (с). Менші значення на

вертикальній осі свідчать про кращу продуктивність алгоритму. Порівняння алгоритму динамічно-наслідкового обмеження переміщення (0,03-0,04 с) і Weighted Score Sort (0,01-0,015 с).

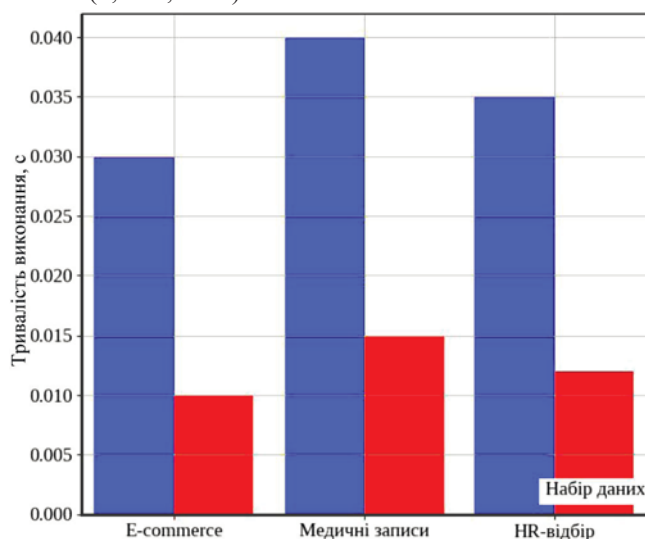


Рис. 4. Графік тривалості виконання сортування структурованих даних / Bar chart of runtime for sorting structured data

Отже, на рис. 4 показано порівняння тривалості виконання (с) алгоритму динамічно-наслідкового обмеження переміщення (сині стовпці) та Weighted Score Sort (червоні стовпці) на наборах даних: E-commerce, Медичні записи і HR-відбір

Результати порівнянь підтверджують, що створена модель алгоритму сортування перевершує алгоритм Weighted Score Sort за стабільністю ($\tau = 0,92 - 0,95$ проти 0,60-0,65) і контролем переміщень (2,1-2,5 проти 5,5-5,8), скорочуючи переміщення елементів на 50-60 % завдяки параметру $R = N/(i+1)$. Незначна різниця у тривалості виконання (0,03-0,04 с проти 0,01-0,015 с) компенсується вищою інтерпретованістю та стабільністю розробленої моделі, що робить її кращою для задач багатокритеріального ранжування даних з високими вимогами до багатокритеріального сортування структурованих даних.

Обговорення результатів дослідження. Для обґрунтування доцільності застосування запропонованої моделі проведено порівняльний аналіз із результатами актуальних наукових публікацій.

У роботі [12] досліджено застосування MCDM у надзвичайних ситуаціях, де ключовими є динамічність і гетерогенність критеріїв. Автори пропонують адаптацію через інтелектуальне переналаштування, що потребує значних обчислювальних ресурсів. Розроблена модель, навпаки, досягає адаптивності через параметр $R = N/(i+1)$, забезпечуючи стабільність ($\tau = 0,91 - 0,95$) без зміни структури, що знижує складність реалізації у практичних системах.

Автори роботи [19] запропонували мультикритеріальну класифікацію запасів, що показала високу ефективність за використання дерев рішень. Проте автори зазначають зниження стабільності за великої кількості параметрів. Розроблена модель забезпечує стабільність ($\tau = 0,91-0,95$) і контроль переміщень (2,1-2,5 позиції), що робить її придатнішою для складних логістичних задач.

Як зазначено у роботі [1], інтегрують MCDM із машинним навчанням для пояснюваного вибору постачальників, але їхній підхід потребує попереднього тре-

нування моделей. Модель, яку ми запропонували, працює без тренування, забезпечуючи результат із високою прозорістю механізму – кожне переміщення елемента визначається через математично контрольований параметр, досягаючи стабільності ($\tau = 0,91 - 0,95$) із середнім переміщенням 2,1-2,5 позиції.

Розроблена у роботі [18] адаптивна консенсусна модель для групового прийняття рішень із лінгвістичними оцінками, забезпечує стабільність сортування структурованих даних, судячи з описаного механізму. Наша модель досягає вищої стабільності ($\tau = 0,91 - 0,95$), працюючи без потреби в узгодженні оцінок між користувачами, що дає змогу автоматизувати процес із кращим контролем переміщень (2,1-2,5 позиції).

Рішення, запропоноване у роботі [11], використовує активне навчання для порогового сортування структурованих даних, що потребує інтерактивного збирання прикладів. Розроблена модель, навпаки, реалізує одноразове ранжування даних без прикладів, тільки на підставі формули R , досягаючи стабільності ($\tau = 0,91 - 0,95$) і швидкості (0,05 с на 1000 об'єктів відповідно до експериментальних результатів у статті автора), що зменшує витрати часу.

Дослідження [16] вказує на потребу в моделях MCDM із кращою стабільністю та контролем переміщень, особливо в контексті електронної комерції та аналітики даних. Розроблена модель відповідає цій вимозі, забезпечуючи контроль переміщень (2,1-2,5 позиції) і стабільність ($\tau = 0,91 - 0,95$), що робить її перспективною для широкого спектра прикладних завдань.

У праці [13] зосереджено на навчанні переваг із часовими критеріями, що потребує значних обчислень. Розроблена модель не спеціалізується на часових критеріях, але підтримує вагову адаптацію через параметр R , досягаючи стабільності ($\tau = 0,91 - 0,95$) без навчання, із часом виконання 0,05 с, що знижує обчислювальну складність і забезпечує ефективність у динамічних умовах.

У роботі [14] запропоновано метод DNMA для агрегації експертних оцінок, який забезпечує об'єктивність, але потребує експертного втручання. Розроблена модель працює автономно, досягаючи стабільності ($\tau = 0,91 - 0,95$) і контролю переміщень (2,1-2,5 позиції), що розширює її застосування в автоматизованих системах.

У дослідженні [20] автори розробили інкрементальний підхід до навчання немонотонних переваг, що потребує уточнення користувачем. Розроблена модель, навпаки, забезпечує стабільність ($\tau = 0,91 - 0,95$) без залучення користувача, зберігаючи повну автоматизацію і контроль переміщень (2,1-2,5 позиції).

Підсумовуючи результати порівняльного аналізу, можна стверджувати, що відомі моделі та методи не забезпечують одночасно стабільність ранжування даних, контроль локальних переміщень і збереження часткового порядку без потреби у тренуванні моделей, залученні експертів чи інтерактивній взаємодії з користувачем. Розроблена модель демонструє поєднання цих властивостей завдяки простому параметру регулювання переміщень (R), що забезпечує досягати високої стабільності ($\tau = 0,91 - 0,95$), точного контролю змін (2,1-2,5 позиції) без потреби в зовнішніх втручаннях.

Отже, внаслідок виконаної роботи можна сформулювати такі наукову новизну та практичну значущість результатів дослідження.

Наукова новизна отриманих результатів дослідження – удосконалено метод багатокритеріального сортування структурованих даних із динамічно-наслідковим обмеженням переміщень, що забезпечує стабільність їх упорядкування без повної перебудови масиву структурованих даних.

Практична значущість результатів дослідження – запропонований метод багатокритеріального сортування структурованих даних можна застосувати в інформаційно-аналітичних системах для оброблення великих даних у електронній комерції, кадровому відборі, а також в експертному оцінюванні, управлінні освітніми й медичними реєстрами, ранжуванні запитів і рекомендаційних сервісах.

Висновки / Conclusion

Розроблено алгоритм адаптивного багатокритеріального сортування даних, який забезпечує збереження часткового їх порядку у масиві відсортованих даних і стабільність отриманих результатів сортування шляхом введення динамічно-наслідкового обмеження на їх переміщення. За результатами проведеного дослідження можна зробити такі основні висновки.

1. Виконаним аналізом літературних джерел встановлено, що відомі алгоритми сортування структурованих даних не зберігають порядку багатокритеріального сортування.
2. Розроблений метод з динамічно-наслідковим обмеженням переміщення сортованих структурованих даних забезпечує врахування пріоритетності критеріїв, виконання багатокритеріального сортування структурованих даних у режимах зростання, спадання та ідентифікації збігу значень.
3. Проведеним порівнянням встановлено, що розроблений метод перевершує відомі за стабільністю ($\tau = 0,92 - 0,95$ проти 0,60-0,65) і контролем переміщень (2,1-2,5 проти 5,5-5,8), скорочуючи переміщення структурованих даних на 50-60 %.

References

1. Abdulla, A., & Baryannis, G. (2024). A hybrid multi-criteria decision-making and machine learning approach for explainable supplier selection. *Supply Chain Analytics*, 2, article ID 100074. <https://doi.org/10.1016/j.sca.2024.100074>
2. Alvarez, P. A., Ishizaka, A., & Martinez, L. (2021). Multiple-criteria decision-making sorting methods: A survey. *Expert Systems with Applications*, 183, article ID 115368. <https://doi.org/10.1016/j.eswa.2021.115368>
3. Belahcene, K., Mousseau, V., Ouerdane, W., Pirlot, M., & Sobrie, O. (2023). Multiple criteria sorting models and methods. *Part I: Survey of the literature*. *4OR*, 21(1), 1–31. <https://doi.org/10.1007/s10288-022-00530-4>
4. Ben Amor, S., Belaid, F., Benkraiem, R., Ramdani, B., & Guesmi, K. (2023). Multi-criteria classification, sorting, and clustering: A bibliometric review and research agenda. *Annals of Operations Research*, 325(2), 771–793. <https://doi.org/10.1007/s10479-022-04986-9>
5. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press. URL: <https://mitpress.mit.edu/9780262046305/introduction-to-algorithms>
6. Greco, S., Ishizaka, A., Tasiou, M., & Torrisi, G. (2021). The ordinal input for cardinal output approach of non-compensatory composite indicators: The PROMETHEE scoring method. *European Journal of Operational Research*, 288(1), 225–246. <https://doi.org/10.1016/j.ejor.2020.05.036>
7. He, R., & McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. *Proceedings of the 25th International Conference on World Wide Web*, pp. 507–517. <https://doi.org/10.1145/2872427.2883037>

8. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95. <https://doi.org/10.1109/MCSE.2007.55>
9. Ishizaka, A., & Resce, G. (2021). Best-Worst PROMETHEE method for evaluating school performance in the OECDs PISA project. *Socio-Economic Planning Sciences*, 73, article ID 100799. <https://doi.org/10.1016/j.seps.2020.100799>
10. Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). *MIMIC-III, a freely accessible critical care database*. *Scientific Data*, 3, article ID 160035. <https://doi.org/10.1038/sdata.2016.35>
11. Kadziński, M., & Ciomek, K. (2021). Active learning strategies for interactive elicitation of assignment examples for threshold-based multiple criteria sorting. *European Journal of Operational Research*, 293(2), 658–680. <https://doi.org/10.1016/j.ejor.2020.12.055>
12. Li, T., Sun, J., & Fei, L. (2025). Application of multiple-criteria decision-making technology in emergency decision-making: Uncertainty, heterogeneity, dynamicity, and interaction. *Mathematics*, 13(5), article ID 731. <https://doi.org/10.3390/math13050731>
13. Li, Y., Guo, M., Kadziński, M., Zhang, Q., & Xu, C. (2025). Data-driven preference learning methods for sorting problems with multiple temporal criteria. *European Journal of Operational Research*, 323(3), 918–937. <https://doi.org/10.1016/j.ejor.2024.12.020>
14. Liao, H., Wu, X., & Xu, Z. (2020). DNMA: A double normalization-based multiple aggregation method for multi-expert multicriteria decision making. *Omega*, 94, article ID 102058. <https://doi.org/10.1016/j.omega.2019.04.001>
15. Liao, H., Xiao, Y., & Wu, Z. (2024). A review of multiple criteria sorting methods: Bibliometrics, characteristics, applications and prospects. *Journal of Control and Decision*, 11(4), 1–13. <https://doi.org/10.1080/23307706.2023.2296120>
16. Liu, J., Kadziński, M., Liao, X., & Mao, X. (2020). Data-driven preference learning methods for value-driven multiple criteria sorting with interacting criteria. *INFORMS Journal on Computing*, 32(4), 855–873. <https://doi.org/10.1287/ijoc.2020.0977>
17. Maloku, F., & Maloku, B. (2024). Analyzing IBM HR data: Employee attrition and performance insights. *Journal of Engineering and Applied Sciences Technology*, 6(8), 1–10. [https://doi.org/10.47363/JEAST/2024\(6\)268](https://doi.org/10.47363/JEAST/2024(6)268)
18. Tian, Z., Xu, F., & Nie, R. (2024). An adaptive consensus model for multi-criteria sorting under linguistic distribution group decision making. *Information Fusion*, 108, article ID 102406. <https://doi.org/10.1016/j.inffus.2024.102406>
19. Zainal Abidin, M. N. A., Bazin, N. E. N., Zebari, D. A., & Rahmat, A. N. (2024). Inventory categorization using multiple criteria classification. *Journal of Soft Computing and Decision Support Systems*, 11(3), 1–10. <https://doi.org/10.37934/jscdm.11.3.110>
20. Zhang, Z., Li, Z., & Yu, W. (2025). Lexicographic optimization-based approaches to learning a representative model for multi-criteria sorting with non-monotonic criteria. *Computers & Operations Research*, 175, article ID 106917. <https://doi.org/10.1016/j.cor.2024.106917>

V. R. Turchak¹, O. V. Ovsyak²

¹ Lviv Polytechnic National University, Lviv, Ukraine

² Ukrainian National Forestry University, Lviv, Ukraine

MODEL OF A MULTICRITERIA DATA SORTING ALGORITHM WITH DYNAMIC-CONSEQUENTIAL MOVEMENT CONSTRAINTS

Multicriteria sorting of structured data, while preserving partial order and ranking stability, represents a key challenge in modern analytical systems used in e-commerce, recommendation services, medical platforms, human resource management, logistics, educational technologies, financial analytics, and large-scale structured data processing for decision support. This study proposes an innovative model of a dynamic-consequential sorting algorithm that enables flexible ranking across diverse data types. The algorithm is based on a modified version of Selection Sort and ensures stable ordering of structured data by primary criteria, even under the influence of secondary factors. This is achieved through clearly defined dynamic movement constraints, taking into account the hierarchy of criteria during ranking. The effect of the movement parameter on maintaining partial order and limiting object displacement across data arrays of different types, including numerical and hybrid datasets, is thoroughly investigated. The developed algorithm demonstrates effective reduction of data movements while maintaining ranking stability amid dynamic shifts in criterion priorities. A correlation between the number of criteria and the degree of ranking stability was identified, enabling optimization of sorting procedures for various applied tasks. The superiority of the proposed algorithm over Multi-Key Sort, Weighted Score Sort, and ELECTRE is confirmed in terms of adaptability, interpretability, and stability. The algorithm flexible prioritization of criteria makes it suitable for applications such processes as product ranking, medical data assessment, logistics optimization, and educational performance analysis. This model is relevant for analytical systems that require adaptive, controlled, and interpretable ranking, offering further potential for optimization in big data processing, real-time system integration, scaling to high-load platforms, and adaptation to domain-specific tasks, such as personalized recommendations, automated decision-making, and financial risk analysis.

Keywords: partial order; ranking stability; criterion prioritization; adaptive sorting of structured data; order preservation.