

КОНСТРУЮВАННЯ АЛГОРИТМІВ СОРТУВАННЯ

Макаров О. В.¹, Шинкаренко В. І.²

¹Український державний університет науки та технологій, аспірант Україна

²Український державний університет науки та технологій,

д.т.н., професор, Україна

Анотація. З розвитком цифрових технологій та збільшенням обсягів оброблюваних даних ефективність алгоритмів сортування набуває критичного значення. У роботі розглянуто еволюцію сортувальних алгоритмів від класичних до гібридних методів, зокрема Timsort та Introsort, які демонструють покращені часові характеристики та стабільність у порівнянні з традиційними підходами. Окрема увага приділена методам передобробки даних та їх впливу на продуктивність.

Запропоновано підхід конструктивно-продукційного моделювання для створення адаптивних алгоритмів сортування, що дозволяє комбінувати існуючі методи та формувати нові ефективні алгоритми. Використання генетичного алгоритму у процесі конструювання дозволяє автоматизувати вибір оптимальних стратегій сортування відповідно до характеристик вхідних даних.

Отримані результати підтверджують перспективність застосування конструктивно-продукційного підходу для побудови адаптивних алгоритмів сортування, що забезпечують високу продуктивність у різних умовах.

Ключові слова: конструктивно-продукційне моделювання, програмне забезпечення, інформаційні технології, алгоритм, формальні граматики, генетичний алгоритм.

Вступ. З розвитком цифрових технологій та глобальної мережі Інтернет, обсяги інформації, що генеруються, передаються та зберігаються, стрімко збільшуються. Задачі пов'язані з обробкою великих масивів даних породжують нові виклики і потребують неординарних та ефективних рішень. У цьому контексті важливість ефективних алгоритмів сортування є надзвичайно високою. Сортування є фундаментальною операцією, що впливає на продуктивність численних процесів, включаючи обробку запитів у базах даних, роботу операційних систем та аналіз великих обсягів інформації.

Історично алгоритми сортування еволюціонували від простих методів, таких як сортування бульбашкою (Bubble Sort) або вставками (Insertion Sort), що мають обчислювальну складність $O(n^2)$, до більш оптимізованих підходів,

придатних для великих масивів даних. Швидке сортування (QuickSort) та сортування злиттям (MergeSort) значно підвищили ефективність обробки даних, досягнувши середньої складності $O(n \log n)$. Розвиток технологій призвів до появи ще досконаліших рішень, таких як гібридні алгоритми. Наприклад, Timsort [1], який поєднує підходи сортування вставками та злиттям, або Introsort, що адаптується до різних ситуацій, комбінуючи швидке, пірамідальне сортування (Heapsort) та сортування вставками.

Гібридні алгоритми поєднують сильні сторони різних підходів для досягнення оптимальної продуктивності. Останнім часом з'явилися дослідження, присвячені застосуванню методів штучного інтелекту та еволюційних обчислень, зокрема генетичних алгоритмів [2], для оптимізації процесів сортування. Генетичні алгоритми дозволяють адаптувати стратегії сортування до конкретних особливостей вхідних даних, що може забезпечити суттєве підвищення ефективності у спеціалізованих випадках.

Окрім вибору відповідного алгоритму сортування, важливу роль відіграє передобробка даних. Детерміновані методи передобробки [3], такі як передбачення позиції елемента відповідно до діапазону значень у масиві або розвертання усіх послідовностей елементів що стоять у зворотному порядку, дозволяють значно скоротити час сортування, якщо вхідні дані мають певну структуру. Стохастичні підходи [4], включаючи випадкове перемішування, використовуються у випадках, коли структура даних невідома або постійно змінюється. Використання стохастичних методів може допомогти уникнути гірших сценаріїв у швидкому сортуванні [5] або покращити адаптивність алгоритмів до різних типів навантаження. Таким чином, розвиток методів передобробки відкриває нові можливості для підвищення продуктивності сортування та ефективності роботи з великими обсягами інформації.

Формальні граматики [6] відіграють важливу роль у створенні та аналізі алгоритмів сортування. Використання граматичних правил дозволяє формально описувати структуру алгоритмів та їх трансформації. Це дає можливість автоматизованого синтезу ефективних алгоритмів сортування, адаптованих до конкретних типів даних або умов використання. Формальні

граматики також застосовуються для побудови оптимізованих послідовностей операцій та перевірки коректності алгоритмів.

Основний матеріал. У контексті алгоритмів сортування формальні граматики можуть бути використані для формалізації структури алгоритмів, автоматичної генерації їх варіантів та аналізу коректності їх роботи.

Генетичний алгоритм використовується для визначення алгоритму сортування найбільш ефективного у заданих умовах. Кожний алгоритм сортування кодується хромосомою. Генами є множина частин відомих алгоритмів сортування, алгоритми передобробки даних а також самостійні відомі алгоритми сортування. Проводяться експерименти за результатами яких відбираються кращі алгоритми, виконуються мутації, створюються нові популяції.

Хромосома представляє собою структуру-дерево [7]. Кожний вузол включає початкову і кінцеву послідовності генів, а також посилання на вузли-нащадки. Для отримання кінцевої послідовності генів виконується обхід дерева у глибину. Таким чином використовується наступний порядок генів:

- початкової послідовності,
- лівого вузла-нащадка,
- правого вузла-нащадка,
- кінцевої послідовності;

Щоб гарантувати сортування масива, кожна початкова послідовність листкового вузла закінчується геном що відповідає одному із існуючих загальновідомих алгоритмів сортування, таких як швидке (quicksort), злиттям (mergesort) або пірамідальне (heapsort) сортування.

Для формування хромосом використовується конструктивно-продукційний підхід. Узагальненим конструктором займається трійка:

$$C = \langle M, \alpha, \Lambda \rangle, \quad (1)$$

Де M – поновлюваний неоднорідний носій, α – сигнатура, яка складається із множини операцій зв'язування, підстановки і вивода, операцій над атрибутами і відношення підстановки, Λ – інформаційне забезпечення конструювання, яке включає призначення, умови початку та завершення конструювання, правила підстановки та обмеження.

Визначимо конкретизацію конструктора що породжує алгоритми сортування. Термінальним алфавітом Σ є гени генетичного алгоритму. Нетермінальний алфавіт $N_1 = \{\phi a_i\}$ складається з багатьох допоміжних символів з атрибутами, що позначаються грецькими літерами.

Доступність правил підстановки визначається значеннями атрибутів. Після виконання підстановки проводяться операції над атрибутами. У розроблених правилах підстановки виникає необхідність виконання операцій над атрибутами як після так і перед операцією підстановки.

Правило $\Psi_1 : \phi a_{s_1}, g_1 \rightarrow \Psi$, включає відносини підстановки s_1 та операції над атрибутами $g_1 : \phi g_{1,1}, g_{1,2} \rightarrow \Psi$. Наприклад правило яке дозволяє додати до хромосоми ген швидкої передобробки даних.

$$s_1 = \langle a\tau_1 \rightarrow QP \cdot a \rangle_{g_{1,1}} = \langle ==(\tau_1, pGU, false) \rangle_{g_{1,2}} = \langle \frac{+(nG, nG, 1)}{:= (pGU, true)} \rangle, \quad (2)$$

де τ_1 – атрибут що визначає доступність правила, QP – ген швидкої передобробки, a – нетермінал, pGU – атрибут який вказує чи була остання виконана операція підстановки із геном передобробки, nG – кількість генів у поточному вузлі дерева-хромосоми. Також визначимо математичні операції використані у формулі: $==(a, b, c)$ – операція порівняння b і c , результат записується у a ; $+(a, b, c)$ – операція додавання b і c , результат записується у a ; $:= (a, b)$ – операція присвоєння, значення b записується у a .

Відносини та операції зв'язування елементів кострукцій для класичної хромосоми генетичного алогоритму що має текстову форму представляють собою конкатенацію символа гена до результуючого рядка. Для хромосоми-дерева новий ген додається до початкової послідовності генів поточного вузла дерева. Певні правила підстановки реалізують додавання нових вузлів-нащадків до поточного вузла. І додавання генів до початкової і кінцевої послідовностей генів кожного вузла-нащадка.

Висновки. Конструктивно-продукційне моделювання застосовується до конструювання алгоритмів сортування як засіб формування нових ефективних алгоритмів на основі існуючих. Використання цього підходу дозволяє аналізувати компоненти алгоритмів, комбінувати їх для отримання нових методів, а також здійснювати адаптацію до специфічних структур даних.

Завдяки конструктивно-продукційному підходу можливо створювати алгоритми, які будуть стійкими до різних умов використання та зможуть самостійно змінювати свою поведінку залежно від характеристик вхідних даних.

ЛІТЕРАТУРА

1. Peters T. Timsort description, accessed june 2015. <http://svn.python.org/projects/python/trunk/Objects/listsort.txt>.
2. David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. – 1989. p. 412.
3. Шинкаренко В. І., Макаров О. В. Дослідження ефективності деяких детермінованих методів передобробки сортування даних. Проблеми програмування – 2023. № 4. С. 3-14. doi: 10.15407/pp2023.04.003
4. Шинкаренко В. І., Дорошенко А. Ю., Яценко О. А., Разносілін В. В., Галанін К. К. Двохкомпонентні алгоритми сортування. Проблеми програмування. — 2022. № 3-4. С. 32-41.
5. Marceau, Robert R. An Analysis of the Worst-Case Performance of Quicksort. Rivier Academic Journal – 2011. № 7.1.
6. Гавриленко С. Ю. Формальні мови, граматики та автомати. Харків: НТУ «ХПІ» – 2021. с. 133.
7. Shinkarenko V. I., Makarov O. V. Structural Adaptation of Sorting Algorithms Based on Constructive Fragments. CEUR-WS. – 2024. № 3806. p. 16-29.

CONSTRUCTION OF SORTING ALGORITHMS

O. Makarov, V. Shynkarenko

Abstract. *With the development of digital technologies and the increase in the volume of processed data, the efficiency of sorting algorithms is becoming critical. The paper considers the evolution of sorting algorithms from classical to hybrid methods, in particular Timsort and Introsort, which demonstrate improved time characteristics and stability compared to traditional approaches. Special attention is paid to data preprocessing methods and their impact on performance.*

A constructive-synthesizing modeling approach is proposed to create adaptive sorting algorithms, which allows combining existing methods and forming new effective algorithms. The use of a genetic algorithm in the design process allows automating the selection of optimal sorting strategies according to the characteristics of the input data.

The results obtained confirm the prospects of using a constructive-synthesizing approach to build adaptive sorting algorithms that provide high performance in various conditions.

Keywords: *constructive-synthesizing modeling, software, information technology, algorithm, formal grammars, genetic algorithm.*

REFERENCE

1. Peters T. Timsort description, accessed june 2015. <http://svn.python.org/projects/python/trunk/Objects/listsort.txt>.
2. David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley. – 1989. p. 412.
3. Shinkarenko V. I., Makarov O. V. Study of the Efficiency of Some Certain Deterministic Preprocessing Methods for Sorting Algorithms. Problems of Programming – 2023. No. 4. pp. 3–14. doi: 10.15407/pp2023.04.003
4. Shinkarenko V. I., Doroshenko A. Yu., Yatsenko O. A., Raznosilin V. V., Galanin K. K., Data Stochastic Preprocessing for Sorting Algorithms, CEUR in: Proceedings of the 13th International Scientific and Practical Programming Conference UkrPROG 2022, vol. 3501, 29- 38.
5. Marceau, Robert R. An Analysis of the Worst-Case Performance of Quicksort. Rivier Academic Journal – 2011. № 7.1.
6. Havrylenko S. Yu. Formal Languages, Grammars, and Automata. Kharkiv: NTU «KhPI» – 2021. p. 133.
7. Shinkarenko V. I., Makarov O. V. Structural Adaptation of Sorting Algorithms Based on Constructive Fragments. CEUR-WS. – 2024. № 3806. p. 16-29.