

ФОРМУВАННЯ АЛГОРИТМІВ СОРТУВАННЯ ЗАСОБАМИ КОНСТРУКТИВНО-ПРОДУКЦІЙНОГО МОДЕЛЮВАННЯ ТА ГЕНЕТИЧНОГО АЛГОРИТМУ

Макаров О. В.¹ [ORCID], Шинкаренко В.І.² [ORCID]

¹Український державний університет науки та технологій, аспірант, Україна

²Український державний університет науки та технологій,

д.т.н., професор, Україна

Анотація. Розглянуто підхід до автоматизованого синтезу алгоритмів сортування на основі конструктивно-продукційного моделювання. Описано конструктивну модель хромосоми деревовидної структури, яка кодує алгоритм сортування у вигляді ієрархічної композиції алгоритмічних фрагментів і допоміжних операцій. Представлено систему з трьох взаємопов'язаних конструкторів: конструктора формування хромосоми-дерева, конструктора-трансформера для перетворення деревоподібної хромосоми у лінійну послідовність генів та конструктора-трансформера, що забезпечує генерацію програмного коду алгоритму сортування мовою програмування.

Показано, що використання чотирьох етапів конструктивно-продукційного моделювання – спеціалізації, інтерпретації, конкретизації та реалізації – дозволяє формалізувати процес переходу від абстрактного опису алгоритму до його виконуваної програмної реалізації. Застосування генетичного алгоритму забезпечує еволюційний відбір та оптимізацію алгоритмів сортування за заданими критеріями якості з урахуванням властивостей вхідних даних і обмежень обчислювального середовища.

Запропонований підхід створює умови для структурної адаптації алгоритмів, поєднання базових алгоритмічних примітивів у нові коректні композиції та розширення простору пошуку ефективних алгоритмічних рішень. Отримані результати підтверджують доцільність використання конструктивно-продукційного моделювання як основи для автоматизованого синтезу та оптимізації алгоритмів сортування.

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

Сучасний етап розвитку інтелектуальних інформаційних технологій характеризується стійким зростанням інтересу до задач автоматизованого синтезу алгоритмів [1], зокрема алгоритмів обробки, впорядкування та аналізу

даних. Алгоритми сортування є базовим компонентом багатьох програмних систем, а їх ефективність безпосередньо впливає на продуктивність прикладних і системних рішень. Традиційні підходи [2] до проектування таких алгоритмів здебільшого спираються на аналітичні міркування та евристичний досвід розробника, що істотно обмежує простір пошуку можливих структур алгоритмів і ускладнює виявлення нетривіальних або вузькоспеціалізованих рішень, адаптованих до конкретних класів задач або даних.

У цьому контексті перспективним напрямом є застосування конструктивно-продукційного моделювання [3], яке дозволяє формалізувати процес формування алгоритмів у вигляді послідовного застосування конструктивних правил і продукцій. Такий підхід забезпечує явне представлення структури алгоритму, підтримує модульність та відкриває можливість систематичного перебору альтернативних варіантів побудови. Поєднання конструктивно-продукційного моделювання з методами еволюційних обчислень, зокрема генетичними алгоритмами [4], створює передумови для автоматизованого формування, оцінювання та оптимізації алгоритмів сортування за заданими критеріями якості.

У межах запропонованого підходу алгоритм сортування подається у вигляді хромосоми-дерева [5], яка кодує його логічну та структурну організацію на рівні конструктивних елементів. За допомогою спеціалізованих конструкторів здійснюється перехід від деревоподібного подання до лінійної генетичної форми, придатної для застосування операторів селекції, схрещування та мутації, а також до програмного коду, що може бути безпосередньо виконаний і протестований. Це дозволяє реалізувати замкнений цикл еволюційного синтезу, в межах якого відбувається автоматизований вибір найкращих варіантів алгоритмів сортування відповідно до заданих показників ефективності, таких як швидкодія, кількість порівнянь або адаптивність до характеристик вхідних даних.

Частини існуючих алгоритмів сортування, а також допоміжні операції, подаються у вигляді символів, що відповідають англійським аббревіатурам (FSS, BSB, QP, SR тощо). У межах конструктивно-продукційного підходу ці

символи інтерпретуються як гени хромосоми генетичного алгоритму та утворюють базовий алфавіт конструктора.

Перший конструктор призначений для формування початкової структурної моделі алгоритму у вигляді хромосоми-дерева. Побудова здійснюється відповідно до заданої системи продукційних правил, що визначають допустимі комбінації генів і порядок їх композиції.

Кожний вузол має початкову і кінцеву послідовності генів. Усі вузли окрім листових мають вказівники на лівий і правий вузли нащадки, що дозволяє відображати ієрархічну структуру алгоритму.

Визначимо спеціалізацію конструктору C_{SA} – моделі формування хромосоми що кодує алгоритм сортування. Для формування конструкцій у вигляді хромосоми у C_{SA} передбачені операції підстановки, часткового та повного виводу. Вивід виконується від початкового нетерміналу, який є початковою формою, ітераційним виконанням операцій часткового виводу. Операція часткового виводу обирає одне з правил з множини Ψ та виконує заміну лівої частини правила на праву у поточній формі та виконує операції над атрибутами. Вона використовує операцію підстановки: саме зміну поточної форми і є частиною операції повного виводу починаючи з початкового нетерміналу і закінчуючи готовою конструкцією.

У процесі формування хромосоми операції підстановки (на основі відповідних відношень підстановки) реалізують додавання терміналів (генів) до вузлів дерева і нових вузлів до дерева

Розглянемо особливості відношень підстановки.

$\beta \xrightarrow{\tau} \gamma \begin{matrix} \nearrow \rho \\ \searrow \delta \end{matrix}$, де β , γ , δ і ρ – деякі послідовності терміналів і нетерміналів.

Відношення надає можливість виконати заміну β на γ у сформованій частині хромосоми, а конкретно у поточному вузлі хромосоми-дерева, якщо в ній є β і атрибут доступності τ дорівнює true. Відношення « \nearrow » і « \searrow » дають можливість додати лівий і правий вузли нащадки до поточного вузла дерева і додати ρ і δ до відповідних вузлів.

Побудована хромосома-дерево є проміжним поданням алгоритму сортування і використовується як основа для подальшого конструктивного перетворення. На наступному етапі за допомогою другого конструктора здійснюється трансляція деревоподібної структури у лінійне генетичне подання.

Визначено C_{CD} – конструктор-трансформер, який реалізує перетворення дерева-хромосоми з закодованими складовими алгоритму сортування у масив (послідовність) генів. Для цього застосовується обхід дерева в глибину (DFS), який забезпечує збереження логічної послідовності виконання алгоритму.

Конструктор C_{CD} містить правила підстановки $\psi_i = ((s_{i,1}, s_{i,2}), (g_{i,1}, g_{i,2}))$, де відношення $s_{i,1}$ реалізує розбір хромосоми з використанням стеку для збереження проміжних результатів. Відношення $s_{i,2}$ призначене для формування списку генів. Операції виконуються у наступній послідовності: спочатку виконуються операції над атрибутами $g_{i,1}$, потім операції підстановки $s_{i,1}$ та $s_{i,2}$, і в кінці – $g_{i,2}$.

Конструктор формування тексту програм C_{PF} (теж конструктор-трансформер) декодує послідовність генів, що кодує алгоритм сортування, у текст програми на обраній мові програмування. Кожному гену відповідає фрагмент коду, а їх композиція формує функціонально завершений алгоритм сортування. Конструктор забезпечує синтаксичну та семантичну коректність згенерованого коду, зокрема узгодження змінних, індексів, умов переходу та меж оброблюваних підмасивів.

Конструктор перетворює лінійну послідовність генів на текст програми сортування. Виконується послідовне декодування нетерміналів у фрагменти програми на обраній мові програмування. Усі термінали $T^* \in T_{PF}$ мають атрибут $text \downarrow T^*$ – відповідна частина тексту програми яка кодується геном.

У результаті формується не тільки виконуваний алгоритм сортування, який може бути безпосередньо скомпільований або інтерпретований та використаний для експериментального оцінювання. А ще і текстове представлення коду усього файлу реалізації (.cpp). Відповідний заголовковий

файл (.h) не змінюється у залежності від хромосоми і створюється заздалегідь. Він включає оголошення функції сортування що конструюється.

Як метод еволюційного пошуку у просторі конструйованих алгоритмів сортування використовується генетичний алгоритм. Усі програми сортування конструйовані на попередніх етапах використовуються для сортування вхідних даних різної структури (випадкові, майже відсортовані, зворотно відсортовані масиви). Відсоток кращих індивідів популяції, які визначаються за часовими показниками, переходить до наступної популяції. Решта індивідів створюються за допомогою операцій схрещування і конструювання нових індивідів. Оператори схрещування застосовуються до хромосом деревовидної структури з урахуванням конструктивних обмежень, що знижує ймовірність появи некоректних або нефункціональних алгоритмів.

Висновки

Була запропонована конструктивна модель хромосоми деревовидної структури яка кодує алгоритм сортування. Розроблена модель дозволяє явно відобразити ієрархічну організацію алгоритму та взаємозв'язки між його фрагментами, що є принципово важливим для керованого структурного синтезу. Викладена спеціалізація та конкретизація конструктора формування хромосом, конструктора-трансформера який перетворює хромосому на лінійну послідовність генів. А також конструктора-трансформера послідовності генів у текст програми сортування на мові програмування.

При структурній адаптації алгоритмів досягається можливість гнучко перебудувати їх внутрішню структуру залежно від властивостей вхідних даних, цільових вимог чи обмежень обчислювального середовища. Такий підхід дозволяє поєднувати фундаментальні алгоритмічні фрагменти у нові структури, що зберігають коректність і водночас покращують часові та ресурсні характеристики.

Запропонований підхід забезпечує поєднання базових алгоритмічних примітивів та фрагментів відомих алгоритмів сортування у нові композиційні структури, які зберігають логічну узгодженість і водночас демонструють покращені часові характеристики. Таким чином, конструктивно-продукційне

моделювання виступає ефективним інструментом розширення простору пошуку алгоритмічних рішень.

ЛІТЕРАТУРА / REFERENCE

1. J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection. – MIT Press, 1992. <https://mitpress.mit.edu/9780262111706/genetic-programming/>
2. R. Sedgewick, K. Wayne. Algorithms. – Addison-Wesley, 2011. <https://algs4.cs.princeton.edu/home/>
3. V.I. Shynkarenko, V.M. Ilman. Constructive-Synthesizing Structures and Their Grammatical Interpretations. I. Generalized Formal Constructive-Synthesizing Structure / Cybernetics and Systems Analysis, Volume 50, Issue 5, pp 655-662. <https://doi.org/10.1007/s10559-014-9655-z>
4. M. Mitchell. An Introduction to Genetic Algorithms. – MIT Press, 1998. <https://mitpress.mit.edu/9780262631853/an-introduction-to-genetic-algorithms/>
5. V. Shynkarenko, O. Makarov. Structural Adaptation of Sorting Algorithms Based on Constructive Fragments / 14th International Scientific and Practical Programming Conference, UkrPROG 2024, CEUR Workshop Proceedings. Vol. 3806, pp. 16–29. <https://ceur-ws.org/Vol-3806/>

SORTING ALGORITHMS FORMATION BY MEANS OF CONSTRUCTIVE- SYNTHESIZING MODELING AND GENETIC ALGORITHM

O. Makarov, V. Shynkarenko

Abstract. *An approach to automated synthesis of sorting algorithms based on constructive-synthesizing modeling is considered. A constructive model of a chromosome with a tree-like structure is described, which encodes the sorting algorithm in the form of a hierarchical composition of algorithmic fragments and auxiliary operations. A system of three interconnected constructors is presented: a constructor for forming a chromosome-tree, a constructor-transformer for transforming a tree-like chromosome into a linear sequence of genes, and a constructor-transformer that provides the generation of the sorting algorithm program code written in a programming language.*

It is shown that the use of four stages of constructive-synthesizing modeling – specialization, interpretation, concretization, and implementation – allows formalization of transition from an abstract description of the algorithm to its executable program implementation process. The use of a genetic algorithm provides evolutionary selection and optimization of sorting algorithms according to specified quality criteria, considering the properties of the input data and the limitations of the computing environment.

The proposed approach creates conditions for structural adaptation of algorithms, combination of basic algorithmic primitives into new correct compositions and expansion of the search space for effective algorithmic solutions. The obtained results confirm the feasibility of using constructive-synthesizing modeling as a basis for automated synthesis and optimization of sorting algorithms.

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