

ЕКСПЕРИМЕНТАЛЬНА ОБЧИСЛЮВАЛЬНА СКЛАДНІСТЬ ФОРМУВАННЯ ГРАФОВИХ ФРАКТАЛІВ НА ОСНОВІ КРИСТАЛІЧНИХ ҐРАТОК

Летучий О. І.¹, Шинкаренко В. І.²

¹Український державний університет науки і технологій, магістр, Україна

²Український державний університет науки і технологій, д.т.н., професор, Україна

Анотація. Визначення експериментальної обчислювальної складності формування графових фракталів дає змогу оцінити та порівняти між собою фрактали, які використовують різні типи кристалічних ґраток. Для моделювання графових фракталів розроблено програму на мові C# з окремим модулем для підрахунку відповідних показників. Для генерації фракталів використовується підхід конструктивно-продукційної моделювання на основі формальних граматик, який складається з низки перетворень та дає гнучкі можливості з налаштування генерації графових фракталів. Обчислювальна складність вираховувалась на основі підрахунку операцій арифметичних, порівняння, присвоєння, переходу під час формування фракталів. Визначені залежності від типу використаної кристалічної ґратки та кількості використаних ітерацій. Встановлено, що асимптотична складність алгоритму формування фрактальних ґраток $O(a^*)$.

Ключові слова: показники; обчислювальна складність, фрактали, граф, програмне забезпечення, інформаційні технології, формальні граматики.

Вступ. Фрактал – це геометрична структура, яка має властивість самоподібності, при зміні масштабу, можна побачити повторювані схожі візерунки або патерни [1]. Фрактали знаходять застосування у фізиці, біології, радіотехніці, інформатики тощо. Як приклад можна навести генерацію складних зображень природних об'єктів, таких як гірські ландшафти [2], дерева, поверхні морів, хмари, вогонь.

Кристалічна ґратка – це впорядкована тривимірна структура, яка описує регулярне розташування атомів, іонів або молекул у кристалі. Вона використовується для пояснення фізичних властивостей твердих тіл, таких як міцність, провідність чи оптичні характеристики, а також для моделювання й аналізу матеріалів у науці та техніці [3-4].

Графові фрактали поєднують властивості фракталів та графів [5-6]. Особливістю таких фракталів є графові вершини, які можуть бути навантажені

координатами у просторі, різними фізичними або хімічними властивостями. Прикладом графових фракталів може виступати кристалічна ґратка.

Експериментальна обчислювальна складність – це галузь дослідження в теорії алгоритмів і обчислювальної математики, яка фокусується на емпіричному вивченні складності алгоритмів та програм шляхом проведення експериментів [7]. На відміну від теоретичного аналізу складності, що базується на математичних оцінках часу чи простору в найгіршому, середньому або найкращому випадках, експериментальна обчислювальна складність вивчає фактичні показники алгоритмів.

Для моделювання фракталів можна використовувати наступні підходи: алгоритмічний, функціонально-алгоритмічний із застосуванням системи ітерованих функцій на основі сукупності стискаючих відображень, афінних автоматів, L-систем. Нами застосовано підхід конструктивно-продукційної моделювання (КПМ) на основі формальних граматик [8-9].

КПМ включає в себе конструктивно-продукційну структуру, яка є послідовністю з трьох складових:

$$C = \langle M, \Sigma, \Lambda \rangle,$$

де M – неоднорідний носій структури, Σ – сигнатура, що складається з безлічі операцій зв'язування, підстановки і висновку, операцій над атрибутами і відносин підстановки, Λ – інформаційне забезпечення, яке включає: онтологію, ціль, правила, обмеження, умови початку та завершення конструювання.

Основна частина. Для реалізації моделювання фракталів за допомогою КПМ та проведення досліджень експериментальної обчислюваної складності написана програма на мові C# в об'єктно-орієнтованому стилі, на фреймворку .NET Framework 4.8.1, див Рисунок 6. Графічний інтерфейс користувача написано у розмітці XAML із виростанням фреймворку WPF.

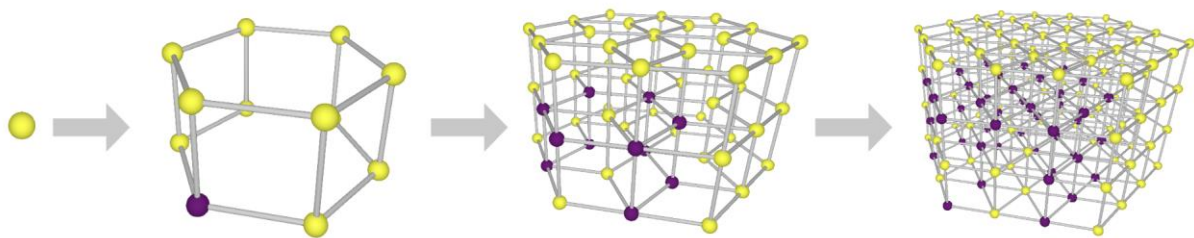


Рисунок 6 – Послідовність сентенційних форм графового фракталу на основі гексагональної кристалічної ґратки

Основні операції, які задіяні для підрахунку показників обчислювальної складності формування мультиатрибутивних просторових графових фракталів: арифметичні (arithmetic), присвоєння (assignment), порівняння (comparison), переходу (jump).

Для цього було переглянута дизасимбльована версія коду на мові C# Рисунок 7, та написана версія алгоритму з додатковими лічильниками операцій.

```

00007FFCDF621BEC mov     rcx,qword ptr [rbp+1A0h]
00007FFCDF621BF3 mov     rcx,qword ptr [rcx+10h]
00007FFCDF621BF7 cmp     dword ptr [rcx],ecx
00007FFCDF621BF9 call    System.Collections.Generic.List`1[[System.__Canon, mscorlib]].Clear() (07FFD3CAE9CB8h)
00007FFCDF621BFE nop

        for (int i = 0; i < Graph.Vertices.Count; i++)
00007FFCDF621BFF xor     ecx,ecx
00007FFCDF621C01 mov     dword ptr [rbp+174h],ecx
00007FFCDF621C07 nop
00007FFCDF621C08 jmp     Crystal.Services.CrystalServiceComponents.Abstraction.Constructor.PerformeSubstitution()+043Dh (07FFCDF621FDDh)
        {
00007FFCDF621C0D nop
            var vertice = Graph.Vertices[i];
00007FFCDF621C0E mov     rcx,qword ptr [rbp+1A0h]
00007FFCDF621C15 call    Crystal.Services.CrystalServiceComponents.Abstraction.Constructor.get_Graph() (07FFCDF2BEF18h)
00007FFCDF621C1A mov     qword ptr [rbp+108h],rax
00007FFCDF621C21 mov     rcx,qword ptr [rbp+108h]
00007FFCDF621C28 mov     r11,offset Pointer to: CLRStub[VSD_LookupStub]@7ffcdef8bb40 (07FFCDEF83298h)
00007FFCDF621C32 cmp     dword ptr [rcx],ecx
00007FFCDF621C34 call    qword ptr [Pointer to: CLRStub[VSD_LookupStub]@7ffcdef8bb40 (07FFCDEF83298h)]
00007FFCDF621C3A mov     qword ptr [rbp+108h],rax
00007FFCDF621C41 mov     rcx,qword ptr [rbp+108h]
00007FFCDF621C48 mov     edx,dword ptr [rbp+174h]

```

Рисунок 7 – Приклад дизасимбльованого коду на мові C# в режимі відладки в середовищі Visual Studio

Порівнюватимуться будуть просторові графові фрактали, які побудовані за допомогою різних типів кристалічних ґраток. Для генерації кожного графового фракталу буде виконуватись десять ітерацій росту фрактала. На основі отриманих даних будуть складені графіки залежності кількості операцій під час виконання кожної ітерації, Рисунок 8. Як приклад наведено експеримент з формування просторового графу з властивостями гексагональної кристалічної ґратки.

Таблица 1

Кількість операцій на кожен крок росту фракталу

№	Arithmetics	Assignments	Comparisons	Jumps	Total
1	547	1155	605	318	2625
2	6017	12235	6410	3348	28010
3	25162	50616	26528	13820	116126
4	55794	112094	58751	30598	257237
5	111041	222721	116741	60774	511277
6	187621	376119	197150	102620	863510
7	298662	598340	313640	163228	1373870
8	440882	883006	462863	240870	2027621
9	627409	1256169	658481	342638	2884697
10	854961	1711451	897146	466804	3930362

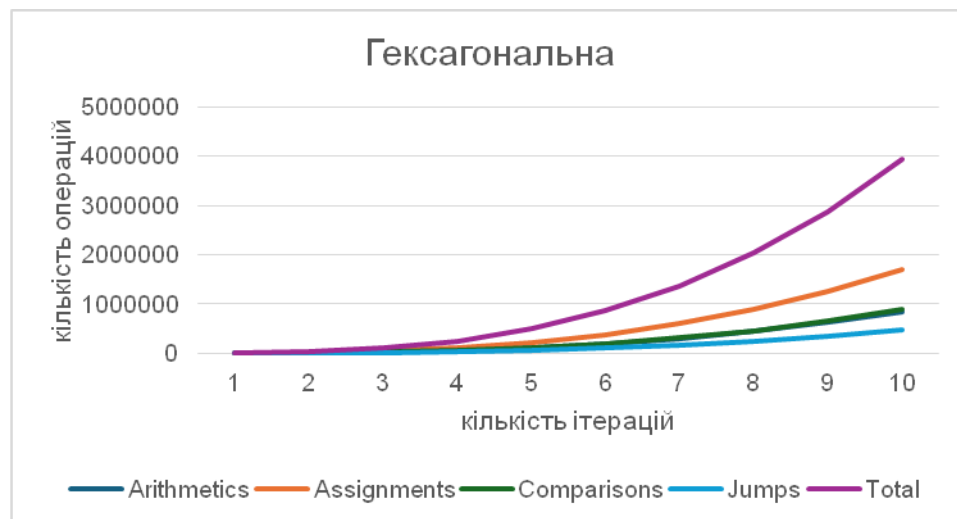


Рисунок 8 – Залежність кількості операцій від кількості ітерацій росту графового фракталу з гексагональним типом кристалічної ґратки

На основі отриманих даних залежності кількості операцій від кількості ітерацій росту графового фракталу виконано розрахунки для знаходження попередньої аналітичної залежності $y = f(x, a, b)$, вид якої є дрібно-раціональний.

Введено заміну змінних $z=1/y$ та $q=1/x$ щоб в площині qOz отримати лінійну залежність. За допомогою методу найменших квадратів знайдемо параметри a та b :

$$b = \frac{n * \sum_{i=1}^n x_i * y_i - \sum_{i=1}^n x_i * \sum_{i=1}^n y_i}{n * \sum_{i=1}^n x_i^2 - (\sum_{i=1}^n x_i)^2} = 0.00322$$

$$a = \frac{\sum_{i=1}^n y_i - b * \sum_{i=1}^n x_i}{n} = -0.00044$$

Кінцева функція має вигляд:

$$y = \frac{x}{0.00039 + 0.00007 * x}$$

Коефіцієнт кореляції $r=0.93$.

Висновок. На основі отриманих даних визначено тип аналітичної залежності. Для визначення параметрів аналітичної залежності використовується метод найменших квадратів, попередньо провівши заміну змінних за потреби. Це надає можливість прогнозування часу формування великих кристалевих ґраток для подальшого їх дослідження.

ЛІТЕРАТУРА / REFERENCES

1. Banerjee S., Hassan M.K., Mukherjee S., Gowrisankar A. Fractal Patterns in Nonlinear Dynamics and Applications, CRC Press. – 2019. №1. P. 35-72. DOI: <https://doi.org/10.1201/9781315151564>
2. Jiang B., Brandt S. A Fractal Perspective on Scale in Geography. ISPRS International Journal of Geo-Information. – 2016. №5(6). DOI: <https://doi.org/10.3390/ijgi5060095>
3. Wang L., Zeng X., Yang H., Lv X., Guo F., Shi Y., Hanif A. Investigation and application of fractal theory in cement-based materials: A review. Fractal and Fractional. – 2021. №5(4). DOI: <https://doi.org/10.3390/fractalfract5040247>
4. Müller U., De La Flor G. Symmetry relationships between crystal structures: applications of crystallographic group theory in crystal chemistry, Oxford University Press. – 2024. №24. P. 243-251.
5. Shynkarenko V., Letuchyi O., Chyhir R. Constructive-synthesizing modeling of fractal crystal lattices, 18th IEEE International Conference on Computer Science and Information Technologies (CSIT). – 2023. DOI: <https://doi.org/10.1109/CSIT61576.2023.10324251>
6. Ille P., Woodrow R. Fractal graphs. Journal of Graph Theory. – 2019. №91(1). P. 53-72. DOI: <https://doi.org/10.1002/jgt.22420>
7. Downey A. Think complexity: complexity science and computational modeling. O'Reilly Media, Inc. – 2018.
8. Skalozub V., Ilman V. Shynkarenko V. Ontological support formation for constructive-synthesizing modeling of information systems development processes, Eastern-European Journal of Enterprise Technologies. – 2018. Vol. 5. №4(95). P. 55-63. DOI: <https://doi.org/10.15587/1729-4061.2018.143968>
9. Shynkarenko V. I. Constructive-Synthesizing Representation of Geometric Fractals. Cybernetics and Systems Analysis. – 2019. №55. P. 186-199. DOI: <https://doi.org/10.1007/s10559-019-00123-w>

GRAPH FRACTALS WITH THE VARIABILITY OF THE FORMATION PROCESS

Oleksandr Letuchyi, Viktor Shynkarenko

Abstract. *The determination of the experimental computational complexity of the formation of graph fractals allows us to evaluate and compare fractals that use different types of crystal lattices. A program in the C# language with a separate module for calculating the corresponding indicators was developed for the modeling of graph fractals. The approach of constructive-synthesizing modeling based on formal grammars is used for the generation of fractals, which consists of a number of transformations and provides flexible options for configuring the generation of graph fractals. The computational complexity was calculated based on the calculation of arithmetic operations, comparisons, assignments, and jumps during the formation of fractals. Dependencies on the type of crystal lattice used and the number of iterations used were determined. It was established that the asymptotic complexity of the algorithm for forming fractal lattices is $O(a^x)$.*

Keywords: *indicators; computational complexity, fractals, graph, software, information technology, formal grammars.*