

УДК 004.421

Підкаура О. А., здобувач

Житомирський державний університет імені Івана Франка

ЗАСТОСУВАННЯ АЛГОРИТМУ IDA* З ВИКОРИСТАННЯМ PATTERN DATABASE ДЛЯ ОПТИМАЛЬНОГО РОЗВ'ЯЗАННЯ КУБИКА РУБІКА

Кубик Рубіка є однією з найвідоміших комбінаторних задач, що належить до класу задач пошуку в просторі станів. Кількість можливих конфігурацій кубика становить приблизно 43 квінтільйони ($4,3 \times 10^{19}$), що робить повний перебір неможливим. Відомо, що будь-який стан кубика можна розв'язати максимум за 20 ходів у метриці FTM (Face Turn Metric) – це так зване «число Бога» (God's Number), доведене у 2010 році. Традиційні методи пошуку в ширину чи A^* потребують величезного обсягу оперативної пам'яті, тому для пошуку оптимальних розв'язків на звичайних комп'ютерах використовують комбінацію алгоритму IDA* (Iterative Deepening A^*) та евристичних баз даних шаблонів (Pattern Databases).

Розроблено повноцінну програму мовою Go під назвою «RubikSolver», що складається з трьох основних модулів: генерація та серіалізація Pattern Database; високошвидкісний рушій виконання ходів та координатного представлення стану; розв'язувач на основі IDA*.

Технічна реалізація «RubikSolver» базується на переході від фізичного представлення кубика (колір кожної грані) до математичного координатного рівня, що дозволяє оперувати індексами в межах Pattern Database. Оскільки повний простір станів у 43 квінтільйони комбінацій неможливо прорахувати прямим перебором, стан кубика розбивається на окремі координати: орієнтацію кутів, орієнтацію ребер та їхні перестановки. Це дозволяє використовувати алгоритм IDA*, який поєднує переваги пошуку в глибину з евристичною оцінкою відстані до розв'язаного стану, що зберігається в попередньо згенерованих базах шаблонів. Ключовою особливістю розробки на мові Go є використання моделі пам'яті та примітивів паралельності для прискорення обчислень. Генерація баз шаблонів відбувається за допомогою горутин (goroutines), які паралельно заповнюють таблиці відстаней, значно скорочуючи час підготовки програми до роботи. Завдяки ефективній серіалізації, ці бази зберігаються у бінарному форматі, що дозволяє програмі швидко завантажувати їх у RAM і знаходити оптимальний розв'язок (до 20 ходів) менш ніж за 15 секунд. Такий підхід забезпечує продуктивність на рівні мов C/C++, зберігаючи при цьому високу читабельність коду та кросплатформність.

Використання симетрій кубика Рубіка є ключовим методом радикального скорочення обсягу даних у Pattern Database. Оскільки куб має 48 власних симетрій (включаючи повороти та віддзеркалення), багато станів є еквівалентними з точки зору кількості ходів до розв'язку. Замість зберігання окремого значення для кожної конфігурації, програма обчислює «канонічний» стан для групи симетричних позицій. Це дозволяє зменшити розмір таблиць у 8–16 разів без втрати точності евристики, що критично для роботи програми на звичайних ноутбуках з обмеженим обсягом оперативної пам'яті.

У процесі пошуку за алгоритмом IDA* симетрія також допомагає уникати зайвих обчислень. Під час генерації баз шаблонів мовою Go, використання симетрій дозволяє швидше заповнювати координатні таблиці, оскільки знаходження відстані для одного стану автоматично визначає відстані для всіх його симетричних копій. Це забезпечує високу швидкість роботи рушія та дозволяє знаходити оптимальний результат у межах 20 ходів за мінімальний час.

Загальний обсяг баз дозволяє запускати програму на звичайному ноутбучі без значних вимог до ресурсів. Середня швидкість розв'язання випадкових позицій є достатньо високою для практичного використання, з середньою довжиною розв'язку, близькою до теоретичного оптимуму. Для позицій, що вимагають максимальної кількості ходів (так званих «суперфліп» та інші екстремальні конфігурації), розв'язок знаходиться в розумні строки. Проведено порівняння з оригінальною реалізацією Kociemba (C) – розроблена програма працює з продуктивністю, близькою до референсної, при значно більшій читабельності та безпеці коду.

Дослідження підтвердило, що використання алгоритму IDA* з базами шаблонів на мові Go дозволяє ефективно знаходити оптимальні розв'язки кубика Рубіка (до 20 ходів) менш ніж за 15 секунд. Завдяки високій продуктивності коду та двофазному підходу програма демонструє стабільну роботу на сучасному обладнанні, залишаючи простір для прискорення через адитивні бази даних та паралельні обчислення.

Список використаних джерел:

1. Kociemba H. Two-Phase-Algorithm and God's Algorithm: God's number is 20. URL: <https://kociemba.org/cube.htm> (дата звернення: 28.11.2025).
2. Rokicki T., Kociemba H., Davidson M., Dethridge J. The Diameter of the Rubik's Cube Group Is Twenty // SIAM Journal on Discrete Mathematics. 2013. Vol. 27, No. 2. P. 1082–1105.