УДК 004.021, 004.023

Д.В. Мальований
О.А. Абдуллаєв
І.В. Богач

ЕВРИСТИЧНІ АЛГОРИТМИ ТА СПОСОБИ ЇХ ЗАСТОСУВАННЯ

Вінницький національний технічний університет

Анотація:
У даній роботі розглянуто поняття і сферу застосування евристичних алгоритмів, а також актуальність їх пристосування під потреби вирішення задач NP-складності задач, а також огляд існуючих концепцій рішень даного класу задач.
Ключові слова: алгоритм, евристичний алгоритм, алгоритмічна обчислювальна складність, оптимізація, квазіоптимальне рішення, недетерміновано-поліноміальна складність.

Abstract:
This paper aims to observe and describe definition and area of application of the heuristic algorithms and relevance of adaption of heuristic algorithms for solving NP-complex problems. Also, existing concepts for solution of given class of problems are described.
Keywords: algorithm, heuristic algorithm, algorithmic computational complexity, optimization, quasi-optimal solution, nondeterministic-polynomial complexity.

Сьогодні відбувається стрімкий розвиток науково-технічного прогресу, а складність задач і завдань зростають значно швидше, ніж доступність і об'єми обчислювальних потужностей. Тому існує потреба у розробці підходів, що дозволять мінімальними втратами в точності в сотні разів пришвидшити пошук рішень до складних задач. Такі підходи в загальному носять назву евристичних алгоритмів. Метою роботи є опис і представлення основних понять і принципів евристичних алгоритмів, а також зрозуміле пояснення як визначення NP-складності, так і необхідності пошуку оптимізованих рішень для задач такої складності. Також проводиться огляд сучасного підходу до пошуку і розробки таких рішень, описаний з певним ступенем деталізації. Деякі з найвідоміших проблем, що підлягають опису, а також декілька серед найпоширеніших у

Загалом, поняття NP складності означає недетерміновано-поліноміальну складність і характеризує оцінку обчислювальної складності та часу, який потрібно витратити на пошук точного рішення задачі за умов, встановлених на початку виконання алгоритму. У сфері дискретної математики, фізики та інших точних наук, як алгоритміка та програмування, оцінку складності слід розуміти, як формульний опис часу, витраченого на пошук рішення, у випадку, якщо вхідні дані наближаються до нескінченності. Фактично така оцінка буде означати кількість вкладених циклів, витрачених програмою на обробку одного екземпляру даних, а геометрично може бути представлена степеневою функцією, наприклад код для виконання виводу в консоль фрази "Hello world!" як суцільної стрічки вважатиметься O (n) = 1 – операцією з константною складністю, в той час, як вивід даної фрази в консоль за допомогою циклу друку масиву літер матиме 'H', 'e', 'l', 'l', 'o', ... має O (n) = n - лінійний час виконання, оскільки з ростом величини вхідної фрази час виконання зростатиме пропорційно, бо програма муситиме вивести в консоль n символів, де n – розмір вхідної фрази. Коли ж справа стосується алгоритмів NP-складності, то процеси їх виконання складаються з такої кількості вбудованих циклів, що їх алгоритмічний час виконання наближається до O (n) = n! або O (n) = e^n, тобто до факторіального або експоненціального. Це означає, що в загальному випадку, час виконання таких алгоритмів (крім частинних випадків) неможливо точно оцінити, і вони можуть зайняти значно більше часу, аніж можливо виділити для того, щоб шукане рішення було актуальним. В реальному житті такі задачі стикаються з двома основними проблемами при їх рішенні:

1) Рішення потребує величезної обчислювальної потужності через велику складність та великі об'єми даних;

2) Рішення повинно бути знайдене в конкретно визначений обсяг часу.

Існують випадки і екземпляри задач, на вирішення яких навіть із застосуванням усіх доступних людству обчислювальних потужностей. Одним із найяскравіших прикладів є пошук усіх можливих перебігів шахової партії. Алгоритм не представляє складності із точки зору програмної імплементації, але через рекурсію і величезну кількість комбінацій і поворотів має експоненціальний час розв'язання. Було доведено.[1] що рішення не може бути повністю обчислене навіть протягом століття і настільки складне, що цього часу буде недостатньо для отримання розв'язку із використанням «досконалого комп'ютера» – обчислювальної машини, що реалізована без жодних втрат у швидкодії і обмежена лише трьома фізичними бар'єрами (світловим, квантовим та термодинамічним) [2].

Існує цілий клас проблем у математиці та фізиці, які за алгоритмічною складністю подібні до описаної шахової задачі. Найбільш відомими та узагальненими видами задач NP-складності, є задача комівояжера та задача китайського листоноші. Значну частину задач NP-складності можна звести до однієї з них.

Формулювання умови обох задач виглядає дуже схожим, хоча має велику різницю. Концептуально, задачу комівояжера можна описати наступним чином:

Існує безліч міст, розміщених у певному координатному просторі, інакше іменованому картою. Мета – обчислити найкоротший шлях, що проходив би через кожне місто, але рівно один раз. Такий шлях називається гамільтоновим. Випадок, коли шлях починається і закінчується в одному місті (вузлі), називається гамільтоновим циклом. Ця задача має на меті знайти відповіді на два основні аспекти:

1) знайти, чи такий шлях може існувати;

2) знайти найкоротший гамільтонів шлях (цикл).

І хоча іноді розв'язання такої задачі являється доволі тривіальним питанням, зокрема на достатньо малих наборах міст, але, у загальному випадку, для отримання найкращого результату доведеться виконати повний перебір усіх можливих гамільтонових шляхів. Не було доведено умови існування найкращого гамільтонового шляху, що одночасно буде найкращим гамільтоновим циклом, так як і не доведено теореми для перевірки гарантованого існування гамільтонового шляху для кожного конкретного даного графу. Єдине, що доведено –достатня умова існування гамільтонового шляху[3]. Фактично це означає, що кожен евклідів граф буде мати гамільтонів шлях.

Представлення задачі китайського листоноші наведено нижче:

Існує карта з набором міст та дорогами (ребрами) між ними. Мета полягає в знаходженні шляху, що проходить через кожне ребро рівно один раз.

У випадках із симетричним та / або неорієнтованим графіком цю задачу можна спростити до детерміновано-поліноміальної складності (клас P, має складність $O(n)=n^m$ із визначеним і скінченним m, де m може бути також і нулем, і одиницею). Якщо сформований граф повно-орієнтований (кожне ребро має напрямок, і лише один), то таку проблему називають NYSSP (New York Street Sweeper Problem, або ж задачею Нью-Йоркського двірника). Змішані графи представляють найбільшу складність для пошуку точного рішення[4].

В результаті, можна дійти висновку, що пошук точних розв'язків таких задач за NP-складність не є достатньо ефективними і тому такі рішення потребують інакшого підходу, аби отримати корисні і релевантні результати.

Навіть після тривалих і наполегливих досліджень, питання можливості розв'язання задач NP-складності із отриманням точного результату за поліноміальний час не було ні доведеним, ані спростованим. Тому пропонований підхід полягає у тому, щоб виконати обчислення набагато швидше, незначно пожертвувавши оптимальністю рішення. Алгоритми, що дають хороший, але не гарантовано найкращий результат, називаються евристичними. Такий результат називається квазіоптимальним. Вони використовують певні моделі поведінки, натхненні або природою, або якимись іншими процесами, і, в свою чергу, поділяються на[5]:

1) Алгоритми із перебудовою шляху;

2) Алгоритми із покращенням шляху;

3) Компоновані алгоритми (об'єднують в собі два попередні).

Вони відрізняються методами побудови та отримання шляху, який вважається фінальним: 1) на кожному кроці додають новий вузол, і таким чином, кількість ітерацій визначатиметься кількістю вузлів-1; 2) На кожній ітерації попередній шлях покращується перестановкою зв'язків між вузлами; and 3) поєднують попередні два підходи.

Одним із найбільш ефективних існуючих алгоритмів – алгоритм Ліна-Кернігана [5]. Але він має достатньо складний спосіб математичної та програмної реалізації. Ще один відомий алгоритмі, який довів свою ефективність – алгоритм еластичної сітки. Загалом, це гібридна модифікація алгоритмів ласо та відпалу. Алгоритм ласо може бути представлений у вигляді кільця вузлів, що охоплює межі області розташування усіх міст (вузлів). Кожну ітерацію це кільце затягується навколо міст і точки на кільці «прилипають» до міст при достатньо близькому розташуванні. Коли кільце максимально затягнуте, і більше не рухаються, отримане рішення вважається кінцевим. Для оптимізації ітерації та затягування використовується алгоритм відпалу. Це уповільнює ласо при запуску і прискорює цільову функцію в кінці, тому знаходять кращі рішення [6].

Однією з ключових особливостей є модифікація кільця ласо таким чином, що воно не обов'язково має охоплювати всі міста на початку, а його радіус може регулюватися у спробах покращити рішення, а також навколо «центру маси» усього набору міст. Також була додана особливість, завдяки якій ласо тепер може не лише затягуватися, але й розтягуватися на деякі міста, сильно віддалені від визначеного центру. Ця особливість значно покращила ефективність та результати для задач і випадків, коли розташування вузлів не підлягає опису з допомогою нормального закону.

Підводячи підсумок, NP-складні проблеми представляють клас проблем, обчислення точного рішення для яких являється недостатньо ефективними, але вони часто зустрічаються в реальному житті. Евристичні або оптимізаційні алгоритми – забезпечують достатньо наближене до найкращого рішення, з використанням обмеженого часу і обчислювальних потужностей. Також можна відслідкувати тенденцію, що з ростом ефективності алгоритму і зменшення алгоритмічної складності, зростає складність його розуміння та втілення.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Hector A. R. Predicting the Outcome of a Chess Game by Statistical and Machine Learning techniques [Електронний ресурс] – Режим доступу: https://pdfs.semanticscholar.org/6115/260e66a8e4e683a43532d8ec9017b4ab6f0f.pdf?_ga=2.257547144.261932811.1587835596-1568095603.1587835596 – Назва з екрану.

2. Bremermann H.J. Quantum Noise and Information, Proc. 5th Berkeley Symp. Math. Statistics and Probability [Електронний ресурс] – Режим доступу: https://projecteuclid.org/download/pdf_1/euclid.bsmsp/1200513783 – Назва з екрану.

3.Li S., Li R., and Feng J. An efficient condition for a graph to be Hamiltonian. - Discrete Applied Mathematics, 2007., vol. 155, no. 14. - 1842–1845 pp.

4. Postman Problem [Електронний ресурс] – Режим доступу: https://www-m9.ma.tum.de/graph-algorithms/directed-chinese-postman/index_en.html – Назва з екрану.

5. Helsgaun K. An Effective Implementation of the Lin-Kernighan Traveling Salesman Heuristic, Department of Computer Science Roskilde University DK-4000 Roskilde, Denmark [Електронний ресурс] – Режим доступу: http://akira.ruc.dk/~keld/research/LKH/LKH-2.0/DOC/LKH_REPORT.pdf – Назва з екрану.

6. Maliovanyi D.V. Elastic Net Algorithm Application to Travelling Salesman Problem [Електронний ресурс] – Режим доступу: https://github.com/str1k6rJP/Java-TSP-Elastic-Net-Solving-Algorithm/blob/precise-optimized/PaperTSPElasticNetUkrainian.docx – Назва з екрану.

Богач Ілона Віталіївна, кандидат технічних наук, доцент кафедри автоматизації та інтелектуальних інформаційних технологій, Вінницький національний технічний університет, ilona.bogach@gmail.com.

Мальований Дмитро Вадимович, студент групи 1ІСТ-18б, Факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, dmytro.maliovanyi@gmail.com.

Абдуллаєв Олексій Алліжанович, студент групи 1ІСТ-18Б, Факультет комп'ютерних систем та автоматики, Вінницький національний технічний університет, fksa.1ict18.aoa@gmail.com.

Bogach Ilona Vitaliivna, PhD, Associate Professor of the department of automation and intelligent information technologies, Vinnytsia National Technical University, ilona.bogach@gmail.com.

Maliovanyi Dmytro Vadymovych, the student of the faculty of computer systems and automation.

Abdullaiev Oleksii Allizhanovych, the student of the faculty of computer systems and automation.