

ТОЧНИЙ АЛГОРИТМ ЗНАХОДЖЕННЯ МАКСИМАЛЬНОЇ КЛІКИ У НЕОРІЄНТОВАНОМУ ГРАФІ

к.т.н. В.Я. Певнев, В.Ю. Ковтун
(подав д.т.н., проф. Ю.В. Стасев)

У статті описується точний алгоритм знаходження максимальної кліки, проведено порівняння з існуючими точними і евристичними алгоритмами. Пропонується можлива реалізація послідовного алгоритму та надані рекомендації по розпаралелюванню наведеного алгоритму.

Використовуючи термінологію [1], клікою (повним підграфом) неорієнтованого графа назвемо такий підграф, у якого будь-які дві вершини зв'язані між собою ребром. Повний підграф назвемо максимальним, якщо він не являється підграфом іншого повного підграфу.

Задача знаходження максимальної кліки належить до найбільш поширених оптимізаційних задач, які інтенсивно досліджуються та мають велике прикладне значення. Пропонується декілька прикладів застосування алгоритму: для рішення задачі розподілу спектру частот, складання оптимальних розкладів, оцінки хроматичного числа графу, для розфарбування графів [1, 2].

Як відомо з [1], задача знаходження максимальної кліки у неорієнтованому графі відноситься до класу NP - повних, тобто для її рішення потрібно виконати експоненційну кількість ітерацій, в залежності від кількості вершин графу та його щільності. На практиці використовують, як евристичні так і точні алгоритми. Кожний з них має свої переваги та недоліки (обмеженість класів графів, на яких алгоритм працює ефективно, час отримання рішення, неможливість ефективного розпаралелення і т.д.).

Відомі підходи до рішення задачі знаходження максимальної кліки основані на послідовному алгоритмі, алгоритмі гілок та границь або алгоритмі з поверненнями. В кожному з цих підходів є свої недоліки та переваги. Для підвищення швидкодії точних алгоритмів використовують різноманітні функції відсікання варіантів під час пошуку, причому хотілося щоб функції відсікання спрацьовували у якомога більшій кількості випадків. Автори статті пропонують свій підхід до проблеми відсікання варіантів.

Порівняємо результати тестування точного алгоритму, що буде описаний далі, та евристичних алгоритмів, що розглянуті у [3, 4]. Експерименти проводились на наборі графів з рівномірним розподілом ребер на комп'ютері з процесором Intel® Pentium® 150 Mhz, 32Mb ОЗП, ОС Windows 95 OSR 2.

Точний алгоритм, що наводиться у статті, складається з двох частин: *евристична*, яка отримує наближену оцінку максимальної кліки у графі та *уточнююча*, яка уточнює отриману квазімаксимальну кліку.

Порівняння точності евристичних алгоритмів В, С [3, 4] з точним алгоритмом А (щільність графів – 0,3) наведено у табл.1.

Таблиця 1

Порівняння точності алгоритмів

Кількість вершин	Алгоритм А	Алгоритм В				Алгоритм С			
		Інтервал зміни значень	Кількість неспівпадань	Максимальна різниця	Кількість неспівпадань з рівнем значимості 0.05	Інтервал зміни значень	Кількість неспівпадань	Максимальна різниця	Кількість неспівпадань з рівнем значимості 0.05
100	6-7	6-7	0	0	0	6-7	0	0	0
200	7-8	7-8	0	0	0	7-8	0	0	0
300	8	7-8	4	1	4	7-8	9	1	9
400	8-9	7-8	1	1	1	7-8	1	1	1
500	8-9	8-9	3	1	3	8-9	11	1	11

Нехай задано граф $G(X, E)$, де X - множина вершин графу, E – множина ребер графу. Евристичний алгоритм В (С) спочатку робить попередню оцінку максимальної кліки у графі - **MaxPower**. Після цього, за допомогою алгоритму швидкого сортування впорядковуються вершини X за спаданням їх ступеня. Вибирається перша вершина x_0 , для неї застосовується функція **FindCliqueExactly**, тобто x_0 буде голова дерева пошуку кліки.

У функції **FindCliqueExactly** будується множина X_0 , до якої ввійшли суміжні для x_0 вершини, видаляються вершини, ступінь яких менша за **MaxPower**. Будується множина X_0' , до якої ввійшли несуміжні вершини з x_0 , видаляються вершини, ступінь яких менша за **MaxPower** [1]. Для вершини x_0 виконується функція **FindCliqueEx**. Серед X_0' видаляються вершини, що мають ступінь меншу за $(MaxPower-2)$ і для кожної вершини, що залишились у X_0' також виконується функція **FindCliqueEx**.

У функції **FindCliqueEx**, для отриманої вершини t , будується множина суміжних вершин T . Серед T береться перша вершина y_0 , для неї будуються множина суміжних вершин Y_0 та несуміжних вершин Y_0' . Проводиться запуск функції **FindCliqueEx** для y_0 і для кожної вершини з Y_0' , причому перевіряється, чи з вершин, що залишились, можливо отримати кліку більшу ніж існуюча. Після кожного запуску **FindCliqueEx** для вершини $y_i \in (Y_0' \cup \{y_0\})$ з T видаляються вершини, згідно теореми 8.3 [1].

Алгоритм (точний) знаходження максимальної кліки у неорієнтованому графі.

Функція Clique ([3], [4]).

Вхід: матриця суміжності графу $G(X, E)$.

Вихід: квазімаксимальна ступінь кліки графу.

1. Початок функції.
2. Дії, згідно [3], [4].
3. Виклик **FindCliqueExactly** для першої вершини у множині X .
4. Кінець функції.

Функція FindClique ([3]).

Вхід: вершина x_i та її ступінь.

Вихід: ступінь кліки, утвореної на множині X_i .

Функція FindCliqueEx.

Вхід: вершина t та її ступінь.

Вихід: ступінь кліки, утвореної на множині T .

1. Початок функції.
2. Якщо кількість суміжних вершин для t дорівнює 1, то збільшення ступеня поточної кліки на 1 та вихід з функції.
3. Якщо не можна отримати кліку більшу існуючої, то вихід з функції.
4. Виділення першої вершини y_0 .
5. Побудова множини Y_0 та Y_0' .
6. Запуск **FindCliqueEx**, для вершини y_0 .
7. Для всіх вершин з Y_0' , що можуть утворити кліку більшу, ніж існуюча, запуск **FindCliqueEx** з подальшим вилученням з T цієї вершини.
8. Кінець функції.

Функція FindCliqueExactly.

Вхід: вершина x та її ступінь.

Вихід: максимальна ступінь кліки, утвореної на множині X .

1. Початок функції.
2. Для вершини x побудова множин суміжних та несуміжних вершин X, X' .
3. Для всіх вершин множини X та X' видалення вершин, ступінь яких менша за (**MaxPower-1**).
4. Для всіх вершин зі $(X' \cup \{x\})$, які можуть утворити кліку більшу існуючої, запуск функції **FindCliqueEx**.
5. Кінець функції.

У алгоритмі, що описаний вище, існує природний паралелізм. Якщо незначно переробити алгоритм, то під час його виконання на багатопроцесорній обчислювальній системі швидкодія значно покращиться. Так, функцію **FindClique** та перший запуск рекурсивної функції **FindCliqueEx** можна виконувати паралельно для різних вершин з $(X_0' \cup \{x_0\})$ і після кожного паралельного кроку виконувати пошук максимального ступеню кліки серед отриманих і т. д. (згідно алгоритму).

Порівняння роботи наведеного алгоритму А з існуючим алгоритмом D [1] надано у табл.2.

Таблиця 2

Порівняння середнього часу знаходження максимальної кліки

Кільк. вершин	Щільність графу – 0.3		Щільність графу – 0.6	
	Алгоритм А, с	Алгоритм D, с	Алгоритм А, с	Алгоритм D, с
100	0.135	1.217	1.9	134.09
200	1.315	20.471	684.42	>6 год.
300	2.54	159.72	10107.17	>24 год.
400	24.4	683.33	-	-
500	200.34	2243.998	-	-

Зробимо висновки з результатів експерименту (табл.2). Результати роботи евристичних алгоритмів, описаних у [3] та [4], за точністю дуже відрізняються від точного в задачах, де наближене рішення не може бути задовільним. або в задачах, де час рішення цільової задачі залежить від точності рішення підзадачі знаходження максимальної кліки.

З таблиці 2 видно, що наведений алгоритм А значно перевищує потужність алгоритму D [1]. При роботі над графами з потужністю, більшою ніж 0.6, значно позначається **NP** - повнота задачі, тобто при збільшенні щільності графу і кількості вершин збільшується глибина регресії функції **FindCliqueEx** та зменшується можливість спрацювання одного з кодів відсічення варіантів (згідно теореми 8.2, [1]). При цьому кількість несуміжних вершин невелика, але збільшується ймовірність спрацювання іншого коду відсічення (по зрівнянню ступеня вершини з оцінкою кліки у графі). Таким чином, алгоритм балансує сам себе в залежності від вхідних даних.

Для підвищення потужності наведеного алгоритму пропонується його виконання на багатопроцесорній ЕОМ з наведеними вище рекомендаціями.

ЛІТЕРАТУРА

1. Део Д., Рейнгольд Д., Нивергельт А. Комбинаторные алгоритмы. Практика и сложность. – М.: Мир, 1987. – 608 с.
2. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи – М.: Мир, 1982. – 416 с.
3. Ковтун В.Ю., Певнев В. Я. Алгоритм знаходження максимальної кліки у графі // Зб. наук. праць ХДПУ. – 1999. – Ч. 3, № 5. – С. 191 - 194.
4. Певнев В. Я., Ковтун В. Ю. Евристичний алгоритм знаходження максимальної кліки у неорієнтованому графі // Системи обробки інформації. Зб. наук. праць. Вип. 1(5). – Харків : НАНУ, ПАНМ, ХВУ, 1999. – С. 66 - 69.