УДК 62. 507

ЕВРИСТИЧНИЙ АЛГОРИТМ ЗНАХОДЖЕННЯ МАКСИМАЛЬНОЇ КЛІКИ У НЕОРІЄНТОВАНОМУ ГРАФІ

к.т.н. В. Я. Пєвнєв, В. Ю. Ковтун*
(подав д.т.н., проф. Ю. В. Стасєв)

У статті описується евристичний алгоритм знаходження максимальної кліки, що може бути реалізований, як у послідовному, так і паралельному варіанті.

Точне рішення задачі знаходження максимальної кліки в неорієнтованому графі завжди потребує таку кількість кроків, що експоненційно зростає із збільшенням кількості вершин у графі. Задача знаходження максимальної кліки входить до основних NP - повних задач [1].

Ясно, що для знаходження рішення на графах з великою кількістю вершин (>300) необхідне використання евристичних алгоритмів, основною рисою яких є велика швидкодія та простота отримання рішення, що у тому, чи іншому випадку менш за все відрізняється від оптимального і рідко дає максимальне відхилення.

Ідеальним виходом було б створення евристичного алгоритму з потребуємою точністю рішення на широкому класі графів і заданою швидкодією. Ці вимоги викликані тими прикладними задачами, які можуть бути розв'язані за допомогою визначення максимальної кліки у графі [1].

Прикладом застосування рішення задачі знаходження максимальної кліки може бути рішення задачі розподілення ресурсів у однорідних багатопроцесорних системах.

Нехай дано граф $\mathbf{G(X,E)}$, де $\mathbf{X}$ – множина вершин, $\mathbf{E}$ – множина ребер, а $\mathbf{X_i} \subseteq \mathbf{X}$ – множина вершин, суміжних з вершиною $\mathbf{x_i}$, відповідно $\mathbf{X'_i = X/\left(X_i \bigcup \{x_i\}\right)}$– множина вершин не суміжних з $\mathbf{x_i}$, $\mathbf{i = \overline{1,n}}$.

Ідея евристичного алгоритму полягає у переборі всіх вершин $\mathbf{x_i}$ графу $\mathbf{G(X, E)}$, у порядку зменшення їх ступеня. Для кожної вершини $\mathbf{x_i}$ виділяється підграф $\mathbf{X_i}$. В $\mathbf{X_i}$ вибирається вершина з найбільшим ступенем $\mathbf{s_i}$, потім з $\mathbf{X_i}$ видаляються всі вершини, які не суміжні з $\mathbf{s_i}$. Цей цикл виконується доки, поки в $\mathbf{X_i}$ не залишиться жодної вершини, при цьому, на кожній ітерації виконується збільшення деякої змінної - максимального ступеня кліки $\mathbf{X_i}$. Евристичність приведеного алгоритму полягає в тому, що у виділеному підграфі $\mathbf{X_i}$ проводиться не повний перебір комбінацій вершин, а ітеративно вибираються вершини з максимальним ступенем, до речі, цей фрагмент залишився і в алгоритмі, що пропонується.

Модифікуємо алгоритм. Для зменшення кількості вершин, для яких потрібно застосовувати функцію знаходження максимальної кліки у підграфі $\mathbf{X_i}$, застосуємо теорему 8.2., що описана у [3].

Нехай **S** – вузол у дереві пошуку **T** (тобто **S** є підмножиною вершин графа **G**, яке індикує повний підграф графа **G**), і нехай перший син вузла **S** у дереві пошуку **T**, який потрібно дослідити, є множиною $\mathbf{S}\cup\{\mathbf{x}\}$ (тобто вершина **x** суміжна з кожною вершиною з **S**). Припустимо, що всі піддерева вузла $\mathbf{S}\cup\{\mathbf{x}\}$ у дереві **T** вже досліджені та породжені всі кліки, що містять в собі $\mathbf{S}\cup\{\mathbf{x}\}$. Тоді необхідно дослідити тільки ті з синів $\mathbf{S}\cup\{\mathbf{v}\}$, які не є суміжними до **x**.

Таким чином, у новому алгоритмі проводиться перебір лише вершин, що не суміжні з вершиною, яка має максимальний ступінь та самої вершини з максимальним ступенем, решта зостається без змін, як і в алгоритмі, що описаний у [2]. У наведеному алгоритмі знаходження максимальної кліки існує природній паралелізм, який значно підвищує швидкодію алгоритму під час його виконання у багатопроцесорній обчислювальній системі. Так, функцію FindClique можливо виконувати паралельно, для різних даних із стеку і після кожного паралельного кроку виконувати пошук максимального ступеня кліки, серед отриманих і т. д. (згідно алгоритму).

Примечание [11]: яка

Алгоритм знаходження максимальної кліки у неорієнтованому графі:

Функція Clique

1. Початок функції.
2. Знаходження ступенів вершин **G**.
3. Знаходження вершини $\mathbf{x_{max}}$ та побудови $\mathbf{X'_{max}}$.
4. Заповнення стеку завдань з множини вершин $\{\mathbf{x_{max}}\}\cup\mathbf{X'_{max}}$ для наступної обробки.
5. Виклик функції FindClique з параметрами - завданнями, що беруться зі стеку: вершина $\{\mathbf{x_i}\}\subset\{\mathbf{x_{max}}\}\cup\mathbf{X'_{max}}$, ступінь вершини $\mathbf{x_i}$.
6. Якщо стек пустий, то перехід до п. 8.
7. Порівняння результату роботи функції FindClique із ступенем максимальної на даний момент кліки. Якщо знайшлася кліка більша, ніж та, що існує, то видаляються з **G** вершини та завдання з вершинами із стеку, ступінь яких менший, ніж ступінь максимальної кліки без одиниці.
8. Перехід до п. 5.
9. Кінець функції.

Функція FindClique

1. Початок функції.
2. Знаходження ступенів вершин $\mathbf{X_i}$.
3. Якщо $|\mathbf{X_i}| > \mathbf{0}$, то перехід до п. 4, інакше перехід до п. 11.
4. Знаходження вершини $\mathbf{s_i}\subset\mathbf{X_i}$, з максимальним ступенем.

5. Побудова $\mathbf{X_i^*}$ з вершинами, що суміжні $\mathbf{s_i}$.

6. Якщо $\left|\mathbf{X_i^*}\right| = \mathbf{1}$, то ступень кліки збільшується на 1, перехід до п. 11, інакше перехід до п. 7.

7. Якщо $\left|\mathbf{X_i^*}\right| = \mathbf{0}$, то перехід до п. 11, інакше перехід до п.8.

8. $\mathbf{X_i^*} \to \mathbf{X_i}$.

9. Знаходження ступенів вершин $\mathbf{X_i}$.

10. Перехід до п. 3.

11. Кінець функції.

Таблиця 1

Порівняння точності евристичних алгоритмів В і А

	Алгоритм А	Алгоритм В
	m_x, с	**m_x**, с
Щільність графу – 0.3		
100	0,086	0,049
200	0,29	0,16
300	0,71	0,44
400	1,45	0,95
500	2,65	1,66
Щільність графу – 0.6		
100	0,17	0,048
200	0,95	0,3
300	2,86	0,95
400	6,49	2,27
500	12,67	4,42
Щільність графу – 0.9		
100	0,92	0,045
200	7,05	0,44
300	23,7	1,6
400	55,87	4,01
500	126,47	7,95

Було проведено порівняння швидкодії та точності алгоритмів, що наведені у цій статті та алгоритму, описаному в [2]. Точчна оцінка математичного очікування часу знаходження максимальної кліки у неорієнтованому графі за допомогою алгоритму А [2] і алгоритму В (розглянуто вище) приведено у табл. 2, з якої видно, що швидкодія алгоритму В значно перевищує швидкодію алгоритму А, з збільшенням потужності графу та його щільності.

Таблиця 2

	Інтервал зміни значення клік (алгоритм А)	Інтервал зміни значення клік (алгоритм В)	Максимальна різниця значень клік між А і В	Кількість неспівпадань	Кількість неспівпадань з рівнем значимості 0.05	Кількість неспівпадань з рівнем значимості 0.1
Щільність графу – 0.3						
100	6-7	6-7	0	0	0	0
200	7-8	7-8	0	0	0	0
300	7-8	7-8	1	5	5	5
400	8-9	8-9	0	0	0	0
500	8-9	8-9	1	8	8	8
Щільність графу – 0.6						
100	11-12	11-12	0	0	0	0
200	13-14	13-14	1	17	17	0
300	14-16	14-16	1	18	18	0
400	15-16	15-16	1	15	15	0
500	16-18	16-18	1	10	10	0
Щільність графу – 0.9						
100	29-32	28-32	2	38	11	0
200	39-42	38-42	3	37	7	0
300	45-49	43-47	3	39	8	0
400	49-53	48-51	4	46	5	0
500	52-55	51-55	3	40	6	0

З таблиці 2 видно, що точність алгоритму В не набагато відрізняється від точності алгоритму А, але з неї видно, що з рівнем значимості 0.05 та 0.1 така точність може бути задовільною.

Таким чином, розроблений евристичний алгоритм визначення максимальної кліки у графі можна використовувати для рішення оптимізаційних задач великої розмірності достатньо точно за припустимий час.

ЛІТЕРАТУРА

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи – М.: Мир, 1982.- 416 с.

2. Ковтун В. Ю., Пєвнєв В. Я. Алгоритм знаходження максимальної кліки у графі // Зб. наук. праць ХДПУ. - 1999. - Ч. 3, № 5. - С. 191 - 194.

3. Део Д., Рейнгольд Д., Нивергельт А. Комбинаторные алгоритмы. Практика и сложность. – М.: Мир, 1987. - 608 с.