УДК 681.3.681.5

НАХОЖДЕНИЕ ЯДЕР ГРАФА С ПОМОЩЬЮ МЕТОДА МАГУ

В. Г. Кобак, В. С. Трощий
Донской государственный технический университет, (г. Ростов-на-Дону, Российская Федерация)

Рассматривается нахождение ядер графа. В качестве алгоритма для нахождения ядер графа используется метод Магу. Нахождение ядра графа методом Магу заключается в том, чтобы найти внутренне и внешне устойчивые множества, которые являются дизъюнктивно нормальными формами логических выражений, построенных при помощи матрицы смежности графа. Исследовалось одно из свойств ядер графа: если задан симметричный граф без петель, тогда любое максимально внутренне устойчивое подмножество является ядром и как потеря симметричности отражается на ядрах.

Ключевые слова: граф, ядро графа, метод Магу, минимально устойчивое множество, максимально устойчивое множество, ориентированный граф.

GRAPH KERNELS FINDING USING THE MAGHOUT METHOD

V. G. Kobak, V. S. Troshchy
Don State Technical University (Rostov-on-Don, Russian Federation)

The paper considers finding of graph kernels. The Maghout method is used as an algorithm for finding the graph kernels. Finding the graph kernel by the Maghout method is to find internally and externally stable sets that are disjunctively normal forms of logical expressions constructed using the adjacency graph matrix. One of the properties of graph kernels has been investigated: if a symmetric graph without loops is given, then any maximum internally stable subset is a kernel, and how the loss of symmetry affects the kernels.

Keywords: graph, graph kernels, Maghout method, minimally stable set, maximally stable set, oriented graph.

Введение. Рассматривая задачу поиска ядер графа, несомненно, стоит упомянуть о её прикладном применении. Несмотря на первый взгляд на довольно малую распространённость, эта задача имеет весьма широкий спектр применений. Так, например, данную задачу можно применять в настольных играх типа шахмат, шашек, и т.д. Эти игры представляются в виде некой абстракции, которую принято называть «игра на графе», где оба игрока своими ходами поочерёдно выбирают вершины до тех пор, пока очередным ходом игрок не зайдет в «тупик» — точку графа, из которой нет выхода. Таким образом, игрок, который постоянно выбирает своим ходом точку, которая входит в подмножество ядер графа, всегда имеет возможность ответить на ход противника таким образом, чтобы самому не оказаться в конечной точке [1].

Основная часть. Задача нахождения ядра графа методом Магу заключается в том, чтобы найти внутренне и внешне устойчивые множества, которые являются дизъюнктивно нормальными формами (ДНФ) логических выражений, построенных при помощи матрицы смежности графа, одинаковые подмножества которых и будут являться ядрами графа.

Математическая постановка задачи. Пусть задан граф $G = (\mathrm{E}, \Gamma)$. Подмножество $S \subset E$ называется *внутренне устойчивым*, если:

$$S \cap \Gamma S = \emptyset. \quad (1)$$

То есть никакие две вершины S не должны быть смежными. Если же $S' \subset S$, то S'— также внутренне устойчивое подмножество.

Максимально внутреннее устойчивое подмножество — это внутренне устойчивое подмножество, не являющееся собственным подмножеством никакого другого внутренне устойчивого подмножества.

Пусть задан граф $G = (\text{E}, \Gamma)$. Подмножество $T \subset E$ называется *внешне устойчивым, если:*

$$(\forall X_i \notin \text{T})\ \text{T} \cap \Gamma X_i \neq \emptyset, \tag{2}$$

т.е. любая вершина X_i, не принадлежащая T, связана по крайней мере с одной вершиной T из другой, начало которой лежит в E–T.

Минимальное внешне устойчивое подмножество — это внешне устойчивое подмножество, не содержащее строго никакого другого внешне устойчивого подмножества.

Пусть задан граф $G = (\text{E}, \Gamma)$. Подмножество $N \subset \text{E}$ называется *ядром графа* G, если N — одновременно внутренне и внешне устойчивое множество, т.е.:

$$(\forall X_i \in N)\ N \cap \Gamma X_i = \emptyset, \tag{3}$$

$$(\forall X_i \notin N)\ N \cap \Gamma X_i \neq \emptyset. \tag{4}$$

Отсюда следует, что ядро содержит всякую вершину X_i с $\Gamma X_i = \emptyset$ и не содержит вершин с петлями. Очевидно, что $\emptyset$ не есть ядро графа. [2]

Методы. Для решения поставленной задачи рассмотрим алгоритм Магу. Он делится на 2 основных этапа:

– нахождение внешне и внутренне устойчивого множества путём построения логического выражения при помощи матрицы смежности и приведение его к ДНФ, выполнив все возможные поглощения;

– нахождение подмножеств, являющихся одновременно внутренне и внешне устойчивыми подмножествами.

Таким образом, получившееся множество, подмножества которого одновременно являются и внешне и внутренне устойчивыми, и будет являться множеством ядер графа.

Рассмотрим пример, который иллюстрирует решение задачи данным алгоритмом. Граф, для которого решается задача, изображен на рис. 1.

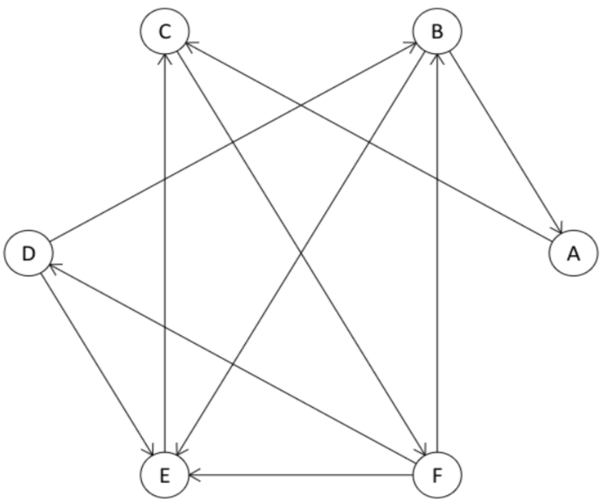

Рис. 1. Ориентированный граф, используемый для примера

Матрица смежности для данного графа изображена на рис. 2.

	a	b	c	d	e	f
a			1			
b	1				1	
c						1
d		1			1	
e			1			
f		1		1	1	

Рис. 2. Матрица смежности графа

Составим выражение для внутренне устойчивого множества: $(A \vee C) \wedge (B \vee A) \wedge (B \vee E) \wedge (C \vee F) \wedge (D \vee B) \wedge (D \vee E) \wedge (E \vee C) \wedge (F \vee B) \wedge (F \vee D) \wedge (F \vee E)$. Приведем его к ДНФ и упростим, выполнив все возможные логические операции поглощения: $(B \wedge C) \vee (A \wedge D) \vee (C \wedge D) \vee (A \wedge E) \vee (A \wedge F)$ — данное выражение является внутренне устойчивым множеством.

Составим выражение для внешне устойчивого множества: $(A \vee C) \wedge (B \vee A \vee E) \wedge (C \vee F) \wedge (D \vee B \vee E) \wedge (E \vee C) \wedge (F \vee B \vee D \vee E)$. Приведем его к ДНФ и упростим, выполнив все возможные логические операции поглощения: $(C \wedge E) \vee (B \wedge C) \vee (A \wedge E \wedge F) \vee (A \wedge C \wedge D)$ — данное выражение является внешне устойчивым множеством.

В получившихся внешне и внутренних устойчивых множествах имеется один общий элемент $(B \wedge C)$, следовательно, множество $\{B, C\}$ является ядром графа.

Вычислительный эксперимент. Одним из свойств ядер графа является следующее: если задан симметричный граф без петель, тогда любое максимально внутренне устойчивое подмножество является ядром [2].

Исследуем, насколько нужно нарушить симметричность графов, убирая из каждой пары дуг по одной так, чтобы у графа не оставалось ядер, а также измерим эффективность работы алгоритма, замерив время его выполнения.

Для вычислительного эксперимента было реализовано программное средство. Для каждого эксперимента было сгенерировано по 50 обыкновенных ориентированных неполных симметричных графов, в каждом из которых случайным образом задавалась матрица смежности. Для каждого эксперимента было взято от 4 до 12 вершин.

На таблице 1 показаны результаты количества вершин, количества сгенерированных дуг среднего количества дуг, которые потребовалось убрать для того, чтобы у графов не осталось ядер.

Таблица 1

Среднее количество удалённых дуг случайно сгенерированных графов

Количество вершин	**Количество дуг графа**	**Среднее количество удалённых дуг**
4	9	4
5	15	7
6	23	10
7	32	14
8	42	18
9	54	22
10	68	29
11	83	34
12	99	41

В таблице 2 показано время, затрачиваемое алгоритмом в среднем для определенного количества вершин.

Таблица 2

Среднее время работы алгоритма

Количество вершин	Среднее время работы (мс)
4	0,23
5	0,44
6	0,82
7	1,54
8	3,12
9	6,38
10	12,13
11	22,28
12	51,44

Заключение. Таким образом, с помощью метода Магу можно найти внутренне и внешне устойчивые множества, одинаковые подмножества которых будут являться ядрами графа.

Библиографический список

1. Домрачеев, Р. Ю. Параллельная реализация поиска ядер графа / Р. Ю. Домрачеев, С. С. Ефимов // Вестник Омского университета. — 2011. — № 2(60). — С. 159–166.

2. Кофман, А. Введение в прикладную комбинаторику / А. Кофман // Москва : Наука, 1975 — С. 180–185.

Об авторах:

Кобак Валерий Григорьевич, профессор кафедры «ПОВТиАС», Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), доктор технических наук, профессор, valera33305@mail.ru

Трощий Владислав Сергеевич, бакалавр кафедры «ПОВТиАС» Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), guttaviva@gmail.com

About the Authors:

Kobak, Valeriy G., Professor of the Department of Computer Engineering and Automated Systems Software, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), Dr. Sci. (Eng.), Professor, valera33305@mail.ru

Troshchiy, Vladislav S., Bachelor’s degree student of the Department of Computer Engineering and Automated Systems Software, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), guttaviva@gmail.com