УДК 519.711

МЕТОД РЕШЕНИЯ ЗАДАЧ "ВЕРШИННОЕ ПОКРЫТИЕ" И "НЕЗАВИСИМОЕ МНОЖЕСТВО"

к.т.н. С.В. Листровой, С.В. Яблочков
(представил проф. А.В. Королев)

В работе предложен алгоритм, перечисляющий все вершинные покрытия и все независимые множества вершин в произвольном графе за время, не превышающее $\mathbf{O(cn^3)}$, где $\mathbf{c}$ – константа, не превышающая $\mathbf{n}$.

Задача определения вершинных покрытий и независимых множеств в произвольных графах является одной из основных задач в математической логике и имеет важное значение в теории построения и управления сложными системами и в теории информации. Поэтому построение эффективных алгоритмов ее решения представляется актуальным.

Пусть задан произвольный граф **G=(V,E)** с множеством вершин $\{v_i\} \in V, i = \overline{(1,n)}$ и ребер **E.** Поставим в соответствие каждому ребру $\{v_i, v_j\} \in E$ графа **G=(V,E)** дизъюнкт $(v_i \vee v_j)$ с двумя переменными. Тогда справедлива следующая теорема.

Теорема 1. Если **f** булева функция, построенная по графу **G=(V,E)** в виде произведения дизъюнктов $(v_i \vee v_j)$ $(\{v_i\} \in \{0,1\}, i = \overline{(1,n)}, j = \overline{(1,n)}, i \neq j)$ и при этом каждый дизъюнкт $(v_i \vee v_j)$ соответствует ребру (v_i, v_j), то все наборы переменных $\{v_i, v_j\}$, на которых она принимает значение «истинно», соответствуют вершинным покрытиям в графе **G=(V,E)**.

Из данной теоремы вытекают два важных следствия.

Следствие1. Для перечисления всех вершинных покрытий графа **G=(V,E)** необходимо определить те системы значений переменных $\{v_i, v_j\}$, при которых высказывание

$$f(V_1, V_2, \ldots V_n) = 1 \tag{1}$$

«истинно».

Чтобы найти эти системы значений переменных $\{v_i, v_j\}$ необходимо привести левую часть (1) к минимальной дизъюнктивной нормальной форме (ДНФ), раскрывая скобки и пользуясь законом поглощения. Такая форма единственна в виду отсутствия в (1) логических отрицаний.

Покажем это на примере графа **G,** приведенного на рис1.

Булева функция для графа будет иметь вид

$$\mathbf{f} = (V_1 v V_2)\ (V_2 v V_3)\ (V_3 v V_4)(V_2\ v V_4)(V_1 v V_4) = \\ = (V_2 v V_1\ V_3)\ (V_4 v V_1 V_2 V_3) = (V_2 V_4 v V_1 V_2 V_3 v V_1 V_3 V_4). \tag{2}$$

Как видно из (2), в результате раскрытия скобок и приведения подобных получаем полный перечень вершинных покрытий графа **G** (рис.1). Ими являются подмножества вершин {2,4}; {1,2,3}; {1,4,3}, а дополнения к ним до полного множества вершин являются независимые множества вершин данного графа {1,3}; {4}; {2}.

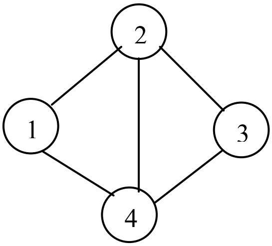

Рис.1. Граф **G**

Следствие 2. Задача перечисления вершинных покрытий в произвольных графах полиномиально разрешима, так как по теореме 2 она сводится к решению задачи «2-выполнимость», которая полиномиально разрешима методом «резолюций» [1].

Воспользуемся особенностью рассматриваемой задачи «2-выполнимость», заключающейся в том, что в ней отсутствуют переменные с логическими отрицаниями. Опишем алгоритм решения данной задачи. Для удобства описания его используем в качестве конструктивной базы алгоритма только ребра и вершины графа.

Алгоритм перечисления вершинных покрытий в произвольных графах. Исходными данными для работы алгоритма может использоваться либо матрица смежности графа или список ребер графа. Предположим, что задан весь список ребер и переменной **X** присвоено значение 1. Тогда алгоритм будет состоять из следующих шагов:

Шаг 1. Удаляем вершину **X** и просматриваем все ребра графа, из тех ребер, в которые она входит. Выписываем номера вершин, формируя множество образующее вершинное покрытие. При этом просмотренные ребра исключаются из дальнейшего анализа.

Шаг 2. Просматривая оставшиеся ребра, выбираем вершину, которая встречается чаще и включаем ее в покрытие, а ребра, в которые она вошла, исключаем из дальнейшего анализа. Если в процессе вычислений имеется несколько ребер, в которые входят по одной различной вершине, формируем несколько подмножеств образующих покрытие, соответствующих числу различных вершин, т. е. в этом случае процесс перечислений покрытий ветвится. Если в процессе вычислений имеется несколько ребер, в которые входят по одной различной вершине. Сформируем несколько подмножеств, образующих покрытие, соответствующих числу различных вершин. В случае, когда имеется несколько вершин с одинаковой частотой появления в ребрах, то в покрытие включаем ту вершину, у которой больше степень.

Шаг 3. Проверяем, все ли ребра исключены из анализа. Если да, то сформированное покрытие запоминается и осуществляется переход к выполнению следующего шага, иначе переходим к выполнению шага 2.

Шаг 4. Проверяем **X>n.** Если да, то в массиве покрытий исключаем избыточные покрытия и алгоритм заканчивает работу. Если нет, то **X:=X+1** и переходим к выполнению шага 1.

Нетрудно видеть, что при одной удаленной вершине алгоритм делает не более $\mathbf{C \cdot n \cdot (n-1)/2}$ операций сравнения и подсчета вершин, а так как вершины удаляются **n** раз, то общая сложность алгоритма не превысит $\mathbf{O(Cn^3)}$, где **C** - константа (**C < n**). Следует отметить, что данный алгоритм позволяет перечислять и все независимые множества произвольного графа, так как дополнение каждого множества вершин покрывающего ребра графа является независимым множеством.

Проблема клики является частным случаем, как проблемы изоморфного подграфа, так и проблемы изоморфной вложимости графа [2]. Поэтому можно ожидать, что предложенный алгоритм окажется полезным и при решении этих задач. Известно, что задача «3-выполнимость» сводится к задаче «вершинное покрытие» [1]. Экспериментальное исследование предложенного алгоритма показало, что если для произвольной индивидуальной задачи «3-выполнимость» [1] строить граф **G=(V,E)** такой, что **G** имеет вершинное покрытие с числом элементов $\mathbf{K} \le \lambda\varsigma\lambda$ тогда и только тогда, когда выполним набор дизъюнкций в индивидуальной задаче «3-выполнимость». Затем, перенумеровав вершины данного графа, с помощью предлагаемого алгоритма перечислить все его вершинные покрытия. Наборы вершин, соответствующие перечисленным покрытиям, определяют наборы переменных (с учетом перенумерации), на которых булево выражение в индивидуальной задаче «3-выполнимость» принимает значение «истинно».

В данной работе показано, что задачи «вершинное покрытие», «максимальное независимое множество», «клика», «3-выполнимость», относящиеся к классу **NP**-полных, могут быть решены за полиномиальное время. Поскольку для основных **NP**-полных задач удалось получить алгоритмы с временной сложностью, не превышающей $\mathbf{O(Cn^3)}$, то можно ожидать, что на основе данных алгоритмов удастся построить эффективные алгоритмы для решения большого класса задач, сводимых к задачам о вершинном покрытии и независимом множестве.

ЛИТЕРАТУРА

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.
2. Емеличев В.А, Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. – М.: Наука, 1990.– 377 с.

Поступила в редколлегию 15.01.2001