УДК 510.62

С. Ю. ШАБАНОВ-КУШНАРЕНКО, И. Ю. ШУБИН, Е. В. ЯВТУШЕНКО

ОБ АЛГОРИТМЕ ВЫРАЖЕНИЯ ОТНОШЕНИЯ ТОЛЕРАНТНОСТИ ЧЕРЕЗ ОТНОШЕНИЕ ЭКВИВАЛЕНТНОСТИ

Пусть имеется множество $T=\{1, 2, ..., n\}$ и бинарное отношение ρ над элементами этого множества. Если для любого $i \in I$ отношение ρ рефлексивно и симметрично для любых $j, k \in I$, то говорят, что на множестве I задано отношение толерантности. Если к тому же имеет место транзитивность, т. е. $\rho(i, j)=1 \,\&\&\, \rho(i, k)=1$ влечет $\rho(i, k)=1$, то говорят, что на множестве I задано отношение эквивалентности.

Моделью отношения толерантности на I может служить неориентированный граф $G(I, R)$ с n вершинами без кратных ребер, в котором каждая вершина имеет петлю, а вершины j и k связаны ребром тогда и только тогда, когда в отношений толерантности $\rho\ (j, k)=1$, этот граф может быть и несвязным.

Моделью отношения эквивалентности на множестве I может служить набор полных графов (с петлями для выражения рефлексивности), где снова вершины j, k связаны ребром, если и только если $\rho(j, k)$. Объединение вершин, входящих в полные подграфы равно I, каждая вершина входит только в один нерасширяемый подграф полный J, т. е. подграфы не соединены между собой.

Имеет смысл рассматривать только те отношения толерантности на I, где соответствующий граф-связный (иначе задача факторизуется и нужно рассматривать каждый фактор отдельно). Условимся для простоты не учитывать петель графа, считая по умолчанию, что они есть (если интерпретировать на графе отношение ρ как «доступность за один шаг», то это все равно, что считать вершину i доступной из самой себя).

После принятия этих условий отношение толерантности не задается неориентированным связным графом без кратных ребер и петель; в дальнейшем будем рассматривать только такие графы.

Рассмотрим следующий пример. Отношение толерантности задано графом, представленным на рисунке. Выразить его как дизъюнкцию отношений эквивалентности так, чтобы все отношения (ребра), принадлежащие G, вошли, по меньшей мере, в одно E. Это можно сделать многими способами, например:

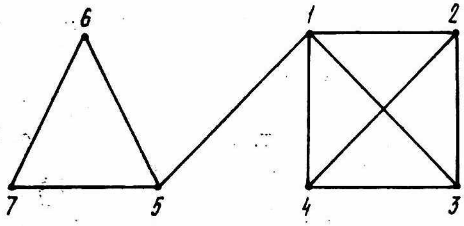
1) $G(1-7)=J(5, 6, 7) \vee J(1,5) \vee J(1, 2, 3) \vee J(1, 2, 3, 4)$;

2) $G(1-7)=J(5, 6, 7) \vee J(1,5) \vee J(2, 3) \vee J(1,3) \vee J(1,3) \vee J(1, 3, 4) \vee J(1,2)$;

3) $G(1-7)=J(5, 6, 7) \vee J(1, 5) \vee J(1, 2, 3, 4)$.

Способы 1 и 2 неэффективны тем, что в сумму справа входят расширяемые клики, и одни и те же ребра повторяются, поэтому в них много слагаемых. Нам желательно найти способ 3, где все клики нерасширяемы и ребра не повторяются.

Таким образом, задача сводится к следующей: дан граф, найти покрытие всех ребер минимальным числом клик.



Граф

Данная задача является NP-полной задачей [1], кроме того, она тесно связана с другой известной NP-полной задачей о нахождении максимальной клики. Поэтому искать простой (полиномальной сложности) алгоритм для точного решения данной задачи не имеет смысла.

Вообще NP-полные задачи все равно приходится решать, и в последенее время для их решения получено достаточно много приближенных алгоритмов, так что NP-полные задачи можно классифицировать по признаку существования алгоритмов их решения, обеспечивающих разумную относительную погрешность

$$\Delta = \frac{|f_p - f_0|}{|f_0|}, \tag{1}$$

где f — функционал, получаемый приближенным алгоритмом — ОПТИМУМ. В этом отношении NP-полные задачи можно классифицировать как плохие, средние и хорошие.

Плохие задачи — это те, для которых не существует простых алгоритмов, которые могли бы обеспечить точность $\Delta \leqslant h$, где h — сколь угодно большая константа. Такой задачей является, например, задача коммивояжера [2].

Средние — это задачи, для которых существуют алгоритмы, гарантирующие точность $\Delta < h$, где h — некоторая константа. Такой является задача о минимальном вершинном покрытии [3], $(h=2)$.

Хорошие задачи — это те, для которых можно указать любое $\varepsilon > 0$ и простой алгоритм, обеспечивающий точность $\Delta \leqslant \varepsilon$. Примером хорошей задачи служит задача о ранце [3].

Известно, что задача о максимальной клике не является хорошей. Гэрри и Джонсон [5] доказали, что если бы задача о максимальной клике была средней, с $h \geqslant 1$, то она была бы хорошей.

Таким образом, решение с точностью $\Delta < 1$ невозможно. Это означает, что задача о максимальной клике является либо плохой, либо средней, но с $h \geqslant 1$. Кроме того, не известно ни одного метода, который бы гарантировал постоянную оценку точности в задаче о покрытии всех ребер графа минимальным числом клик.

Поскольку нет никаких алгоритмов, точность которых оценивалась бы константой, а не возрастающей функцией от n, мы применим для решения нашей задачи алгоритм, являющийся обобщением алгоритма, описанного в [6].

Для описания алгоритма введем некоторые обозначения. Подграф графа $G(I, R)$, который содержит вершины из множества J, $J \subseteq I$, и все ребра из R, инцендентные вершинам из J, будем обозначать $G(J)$. $\Gamma_i(J)$ — это множество всех вершин, соседних с i в графе $G(J)$ (соседними являются две вершины, если они соединены ребром). Через $R(C)$ будем обозначать все ребра, образующие клику $G(J)$.

Алгоритм: A_1:

1°. $J:=I$, $m:=|R|$ (где $|R|$ — обозначает число элементов в множестве R).

2°. $C:=\varnothing$.

3°. Найти в $G(J)$ вершину максимальной степени (степень вершины — количество инциндентных ей ребер). Пусть это j_0.

4°. $C:=c \cup \{j_0\}$, $J:=\Gamma_i(c)$, если $J=\varnothing$, то перейти к шагу 3°.

5°. Выдать C; $\gamma := |c|$; $m:=m-m-\gamma(\gamma-1)/2$, если $m=0$, то конец.

6°. $R:=R/R(C)$; $J:=J/J(C)$ {множество всех изолированных вершин}, перейти к 2°.

Алгоритм неплохо зарекомендовал себя в практических вычислениях. Он реализован на микроЭВМ.

Замечание. В примере решения задачи для графа, изображенного на рисунке, если бы на шаге 3° первого прохода была выбрана вершина 5, то алгоритм выдал бы те же клики, но в порядке {1, 5}→{1, 2, 3, 4}—{5, 6, 7}. Если решать этим алгоритмом (обрывая его на выдаче C, шаг 5°) задачу о нахождении максимальной клики, то можно было бы получить клику либо из четырех вершин, либо из двух. Но при решении поставленной нами задачи склонность алгоритма к локальным ошибкам сказывается в меньшей степени.

Список литературы: 1. *Тери М., Джонсон Д.* Вычислительные машины и труднорешаемые задачи. М., 1982. 215 с. 2. *Пападимитру Х., Стайглиц К.* Комбинаторная оптимизация: Алгоритмы и сложность. М., 1985. 170 с. 3. *Sahni S., Gonrales T.* P-Complete Approximation Problems. ASM. 1976. № 23. P. 555—565. 4. *Ibarra O. H., Kim C. E.* Fast Approximation Algorithms for the Knapsak and Sum of Subset Problems. ASM. 1975. № 22. P. 463—468. 5. *Garly M., Johnos D.* Ihe Complety of Near-Optimal Graph Coloring. ASM. 1978. № 25. P. 499—508. 6. *Johnson D.* Approximation Algorithms for Combinatorial Problems. I. Comp. Systems. 1974. № 9. P. 256—278.

Поступила в редколлегию 23.01.89