Maximum Acyclic Subgraph

Никита Ласков

15 января 2022 г.

Содержание

Введение

Курсовой проект по теории алгоритмов.

1 Основная часть

1.1 Формулировка задачи распознавания, доказательство ее NP-полноты

1.1.1 Формулировка задачи распознавания

Задача 1 (Задача MAS)
Дан конечный ориентированный граф $D = (V, A)$ и константа $B \in \mathbb{N}$. Существует ли подмножество $A' \in A$, такое, что подграф $D = (V, A')$ не содержит циклов и $|A'| >= B$.

1.1.2 Доказательство NP-полноты

Чтобы показать, что задача MAS является NP-полной, требуется:

1. Показать, что $MAS \in NP$

2. Свести к MAS другую известную задачу, чья NP-полнота уже установлена

Определение 1
Задача распознавания P принадлежит классу NP при схеме кодирования c, если $L\left[P, c\right] \in NP$

Определение 2
Язык L принадлежит классу NP, если существует НМТ M, распознающая L, и многочлен $p \in \mathbb{Z}[x]$, такие, что время работы M на любом входе $x \in \Sigma^*$ не превосходит $p(|x|)$

Таким образом, чтобы доказать что задача MAS является NP полной, нам достаточно убедиться в существовании недетерминированной машины тьюринга, которая бы распознавала язык этой задачи.

Действительно, в качестве подсказки достаточно взять набор нулей и единиц длины $|V|$, где каждое значение соответствует конкретной вершине $v \in V$, и единицы стоят на местах вершин, которые входят в максимальный ациклический подграф, а на местах оставшихся вершин - нули. Для этого предлагаю свести к задаче MAS задачу о независимом множестве.

Определение 3
G - конечный граф. $W \in V(G)$ - независимое множество, если
$\forall u,v \in W(uv \notin E(G))$

Задача 2 (О независимом множестве)
Дан конечный неориентированный граф G и число $B \in \mathbb{N}$. Есть ли в G независимое множество размера не менее B.

Преобразуем неориентированный граф из зачачи о независимом множестве G к ориентированному D следующим образом:

$$V(D) = V(G)$$
$$A(D) = \{\{uv, vu\} \mid \forall uv \in E(G)\}$$

Таким образом, мы строим граф на тех же вершинах, и для каждого ребра исходного графа добавляем две разнонаправленные дуги в наш новый ориентированный граф.

При таком построении, если мы найдем в графе G независимое множество W, то мы также нашли бы максимальный ациклический подграф в D. Это правда, так как добавление любой из оставшихся вершин в подграф появился бы цикл, так как в графе G добавленная вершина была бы связана с одной или несколькими вершинами из независимого множества W.

1.2 Частные случаи

Определение 4
Степень вершины графа - количество ребер, инцидентных этой вершине.

Определение 5
Регулярный граф степени k - это граф, все вершины которого имеют степень k.

1.2.1 Регулярный граф степени < 3

Самый простой частный случай - если мы имеем дело с регулярным графом степени меньше 3. Если в таком графе и есть цикл - то это цикл, в который входят все вершины. Чтобы это проверить, достаточно пройтись по всем ребрам, что можно сделать за полиномиальное время.

1.2.2 Регулярный граф степени 3

Если же регулярный граф имеет степень 3 - то все не так однозначно. В общем случае - задача MAS остается NP трудной для таких графов, но для некоторых особых ситуаций мы можем найти приближение с точностью как минимум $\frac{8}{9}$, или даже точное решение за полиномиальное время. [1]

Пусть дан регулярный граф степени 3: $G = (V, E)$, для которого мы хотим найти максимальный ациклический подграф $S \subseteq E$. Будем проводить рассуждения в рамках следующих предположений.

Определение 6
Длина цикла - число ребер, входящих в цикл.

Определение 7
Положительная (отрицательная) степень вершины графа - это число всех исходящих (входящих) ребер. Обозначения: $d^+\{v\}, (d^-\{v\})$

Предположение 1.2.2.1
Все вершины в графе G имеют положительную и отрицательную степени не меньше 1 и суммарную степень 3.

Доказательство. Если в G содержится вершина, у которой положительная или отрицательная степень равна нулю, то мы можем сразу включить все смежные с ней ребра в S, так как они будут содержаться в любом максимальном ациклическом подграфе. □

Предположение 1.2.2.2
Граф G не содержит циклов длины 2 и 3.

Доказательство. Если мы имеем дело с неориентированными циклами, то мы можем договориться для каждого такого цикла включать в S все ребра цикла, при этом сами циклы удалить из рассмотрения. Удаление цикла длины три не добавит новых циклов, так как мы работаем в рамках предположения 1.2.2.1. В случае ориентированных циклов длины 2 и 3, нам достаточно не включать в S какое-то одно ребро цикла. □

Определение 8
α-ребро - ребро (i,j), такое, что

$$d^{-}\{i\} = 2, d^{+}\{i\} = 1 \qquad d^{-}\{j\} = 1, d^{+}\{j\} = 2$$

По лемме 2.1 из [1], если регулярный граф степени 3 не содержит α-ребер, то все циклы в нем не содержат общих ребер. Это утрверждение легко доказывается от обратного. Если мы рассматриваем два цикла в регулярном графе степени три без α-ребер, то возможны два случая: они пересекаются по одному ребру или содержат несколько общих ребер. В случае пересечерния по одному ребру очевидно, что это ребро пересечения обязано быть α-ребром. Если же пересечение состоит из нескольких ребер, то среди этих ребер всегда найдется α-ребро. Это легко проверить следующим алгоритмом. Предположим, что у нас направление движения по общему пути для двух циклов - сверху вниз (для определенности). Мы будем пробовать привести такой пример графа, в котором этот самый общий путь не содержал бы α-ребер. Зная, что все вершины у нас степени 3, и так как это часть цикла, попробуем добавить ребра к вершинам которые находятся внутри пути так, чтобы не появилось α-ребер. Будем идти сверху вниз. При таком подходе мы чередуем сначала добавление ребра, которое ... (Нарисовать картинки, дописать доказательство)

Если в графе нет α-ребер, то мы можем найти максимальный ациклический подграф за полиномиальное время следующим алгоритмом. Мы просто находим цикл в графе, выбрасываем произвольное ребро из этого цикла, остальные добавляем в максимальный ациклический подграф. Также после этого мы стягиваем соответствующие ребра в оставшемся графе таким образом, чтобы 1.2.2.1 и ?? оставались истинными.

Если в графе есть α-ребра, то делаем следующее.

1. Находим α-ребро e в графе
2. Удаляем e Добавляем все $E(e)$ и вершины с нулевой положительной/отрицательной степенью в решение.
3. стягиваем вершины у которых $d^{+}\{v\} = 1$, $d^{-}\{v\} = 1$

Добавить, что такое стягивание вершин.

Понятно, что последний приведенный алгоритм будет давать апроксимацию $\frac{8}{9}$ в регулярном графе степени 3. Если мы нашли α-ребро е для которого |С(е)| = 9 то мы решаем эту компоненту точно, а если нет - то с точностью 8/9, так как для каждого альфа-ребра на удаление одного ребра мы добавляем в наше решение как минимум 8 ребер.

В [1] также показано, что если более аккуратно выбирать α-ребра, то мы можем получить алгоритм с точностью $\frac{11}{12}$.

1.3 Точный экспоненциальный алгоритм

1.3.1 Описание алгоритма

Рассмотрим теперь точный экспоненциальный алгоритм, для этого вначале приведем несколько определений.

Покажем как построить алгоритм для двойственной к MAS задаче:

Задача 3 (FAS)
Дан конечный ориентированный граф $G = (V, A)$. Нужно найти такое подмножество $F \subseteq A$, такое чтобы G - F Был бы ациклическим графом и при этом $|F|$ была минимальной из всех возможных.

Понятно, что если мы знаем решение для задачи FAS, то мы легко найдем и сам максимальный ациклический подграф в исходном графе, как $G\,[A - F]$

Также введем функцию, которая упорядочивает вершины в графе:

Определение 9
$\pi : V \to 1, ..., |V|$ - функция, которая задает некоторую нумерацию на множестве вершин V. При этом, $\forall a = (i, j) \in A$:

$$\pi(i) < \pi(j) \Rightarrow a - forward$$
$$\pi(i) > \pi(j) \Rightarrow a - backward$$

С учетом функции π можно немного переформулировать задачу FAS следующим образом:

Задача 4 (FAS)
Дан ориентированный граф $G(V, A)$. Найти такую перестановку π, что $\sum\limits_{((u,v)\in A\ \&\ \pi(u)>\pi(v))} 1$ - минимальна. Иными словами, требуется найти такую перестановку, в которой количество обратных дуг минимально.

Под оптимальной перестановкой будем понимать такую перестановку, в которой число обратных дуг минимально.

Определение 10
$X(S)$ - число обратных дуг в оптимальной перестановке для индуцированного графа $G[S]$, $S \subseteq V$.

Также $X(S)$ можно представить в виде следующей рекурсивной формулы:

$$X(S) = \min_{u\in S} \left\{ X(S - u) + \sum_{((u,v)\in A\ \&\ v\in(S-u))} \right\} \tag{1}$$

Покажем корректность рекурсивной формулы (1). Важно понимать, что при каждом увеличении размерности, к примеру при переходе от $S - u$ к S, мы при каждом рассмотрении новой вершины u присваиваем есть новый, самый большой номер в нумерации вершин. Соответственно, все дуги, которые приходят из вершин множества $S-u$ в вершину u будут являться обратными по определению. Ну и чтобы найти минимальное количество обратных дуг мы ищем минимум суммы минимального числа обратных дуг для множества без рассматриваемой вершины и числа всех дуг, которые приходят в рассматриваемую вершину из вершин $S-u$.

Перейдем теперь непосредственно к алгоритму [2]. Ключевой структурой данных, которую мы будем использовать, будет массив Y размерности $2^n \times 2$. В ходе работы алгоритма будем рассматривать подмножества множества вершин $S \subseteq V$, и $\forall S$:

$$Y[S,1] = X(S)$$
$$Y[S,2] = \{v | v \in V : X(S) \text{ } is \text{ } minimized \text{ } in \text{ } eq. \text{ } (1)\}$$

Algorithm 1: FAS

Input: A directed graph G
Output: Size of a minimum feedback set

Let Y be a $2^n \times 2$ array indexed from 0 to $2^n - 1$
Initialize $Y[S,1] = \infty$, $Y[S,2] = 0$ for all subsets S $\subseteq V$ *and* $S \neq \varnothing$
Initialize Y$[\varnothing, 1] = Y[\varnothing, 2] = 0$
for $S \subseteq V$ *enumerated in increasing order of cardinality* **do**
    for $u \in V - S$ **do**
        $P = Y[S,1] + \sum_{((u,v)\in A \ \& \ v\in(S-u))} 1$
        if $P = Y[S \cup \{u\}]$ **then**
            $Y[S \cup \{u\}, 2] = Y[S \cup \{u\}, 2] \cup \{u\}$
        if $P < Y[S \cup \{u\}]$ **then**
            $Y[S \cup \{u\}, 1] = P$
            $Y[S \cup \{u\}, 2] = u$
return *Y[V, 1]*

Таким образом в результате работы алгоритма мы получим:

$$Y[V,1] - \textit{число обратных ребер в оптимальной перестановке для } G$$
$$\bigcup_{S\subseteq V} Y[S,2][1] - \textit{множество вершин, которые инцидентны обратным дугам}$$
$$\textit{в минимальном по размеру множестве } F$$

1.3.2 Оценка сложности

Теорема 1.1. *$G(V,E)$ - ориентированный граф, $|V| = n$, $|A| = m$. Тогда размер множества обратных дуг может быть найден за время $O^*(2^n)$ и с использованием $O^*(2^n)$ памяти.*

Доказательство. В алгоритме мы для каждого подмножества S отрабатываем некоторое количество вершин за время $O^*(n)$.

1.4 Полиномиальный приближенный алгоритм

Ссылки: [3], [4]

2 Дополнительные исследования

2.1 Нижние оценки погрешности для приближенных алгоритмов

2.2 Вероятностные алгоритмы

Алгоритм отсюда: [4]

2.3 Исследование вариаций формулировки

Ссылки: [5]

Список литературы

[1] Alantha Newman. The maximum acyclic subgraph problem and degree-3 graphs. pages 147–158, 01 2001.

[2] Venkatesh Raman and Saket Saurabh. Improved fixed parameter tractable algorithms for two “edge” problems: Maxcut and maxdag. *Information Processing Letters*, 104:65–72, 10 2007.

[3] Alantha Newman. Approximating the maximum acyclic subgraph. 05 2014.

[4] Refael Hassin and Shlomi Rubinstein. Approximations for the maximum acyclic subgraph problem. *Information Processing Letters*, 51:133–140, 08 1994.

[5] Mourad Baïou and Francisco Barahona. Maximum weighted induced bipartite subgraphs and acyclic subgraphs of planar cubic graphs. *SIAM Journal on Discrete Mathematics*, 30:1290–1301, 01 2016.