

ЗАДАЧА ШТЕЙНЕРА ДЛЯ АЦИКЛИЧЕСКОГО ГРАФА

Ильченко А. В.

ТАВРИЧЕСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМ. В.И. ВЕРНАДСКОГО
ФАКУЛЬТЕТ МАТЕМАТИКИ И ИНФОРМАТИКИ
ПР-Т ВЕРНАДСКОГО, 4, Г. СИМФЕРОПОЛЬ, 95007, УКРАИНА

Abstract

The Steiner problem for graph without cycles is considered in the paper. The algebraic approach is used for the problem investigation and solution. The operations under simple chains sets are introduced and the properties of this operations are considered. The notions of completeness, base, canonic base for simple chains sets are defined. It is offered a simple algorithm for problem solution which uses previously defined base chains. It is shown the polynomial complexity of the problem.

Введение

Пусть $G = \langle V_G, E_G \rangle$ — конечный связный неориентированный граф. V_G — множество вершин, E_G — множество ребер графа G .

Формулировка задачи Штейнера [1]: В конечном связном взвешенном неориентированном графе G выделено подмножество вершин $U \subseteq V_G$. Требуется найти связный подграф $T = \langle V_T, E_T \rangle$, удовлетворяющий следующим двум условиям:

1. Множество V_T содержит заданное множество U ($U \subseteq V_T$);
2. Граф T имеет минимальный вес среди всех подграфов, удовлетворяющих условию 1.

Очевидно, что искомый подграф является деревом. Он называется деревом Штейнера.

Далее рассмотрена задача Штейнера для ациклического графа. В этом случае перестает играть роль взвешенность графа.

Содержательно, алгоритм построения дерева Штейнера для этого случая состоит из следующих шагов:

1. Выбирается произвольная вершина из заданного множества вершин U . Подграф, состоящий только из этой вершины, объявляется построенной частью дерева Штейнера.
2. Выбирается «ближайшая» к построенной части вершина из заданного множества вершин U , еще не включенная в построенную часть дерева Штейнера. Находится простая цепочка, соединяющая выбранную вершину из U с «ближайшей» вершиной из построенной части дерева Штейнера. Ребра и вершины этой цепочки включаются в построенную часть дерева Штейнера. Полученный подграф объявляется построенной частью дерева Штейнера.

Шаг 2 алгоритма повторяется до тех пор, пока все вершины из U не будут включены в построенную часть дерева Штейнера.

Для реализации этого алгоритма необходимо уметь определять «ближайшую» к выбранной вершине, и строить простые цепочки, соединяющие указанную пару вершин графа. Отметим, что для ациклического графа, выбор вершины, «ближайшую» к построенной части, не играет роли. В качестве «ближайшая» к построенной части можно брать любую вершину из заданного множества вершин U , еще не включенную в построенную часть дерева Штейнера.

Ниже рассмотрен алгебраический подход к решению этой задачи.

1. ЦЕПОЧКИ, ПОРОЖДАЕМЫЕ РЕБРАМИ ГРАФА

Пусть $G = \langle V_G, E_G \rangle$ — конечный связный неориентированный ациклический граф. $V_G = \{\nu_1, \nu_2, \dots, \nu_n\}$ — множество вершин, $E_G = \{e_1, e_2, \dots, e_m\}$ — множество ребер графа G . Цепочка из вершин и ребер графа называется простой, если она не содержит повторяющихся вершин.

Определение 1. Две простые цепочки называются смежными, если у них есть хотя бы одна общая концевая вершина.

Для смежных цепочек определена операция композиции [1]. Композиция двух простых смежных цепочек — маршрут. Удалив «лишний» участок маршрута, получим простую цепочку. В случае связного ациклического графа, «лишний» участок — это подцепочка, проходимая дважды.

Простая цепочка — может быть задана последовательностью ребер, т. е.

- а) множеством ребер, составляющих подцепочку;
- б) порядком этих ребер в цепочке.

Поиск простой цепочки можно проводить в два этапа:

- а) найти ребра, составляющие простую цепочку;
- б) упорядочить найденные ребра в цепочку.

Для поиска ребер, составляющих простую цепочку, соединяющую две заданные вершины, удобно использовать теоретико-множественные операции. Пусть ρ_1, ρ_2 — две смежные цепочки, несовпадающие концевые вершины которых — это те две вершины, которые надо соединить простой цепочкой.

$E_1 = \{e_i\}$ — множество ребер цепочки ρ_1 ,

$E_2 = \{e_j\}$ — множество ребер цепочки ρ_2 .

ПРАВИЛО ВЫЧИСЛЕНИЯ ПРОСТОЙ ЦЕПОЧКИ:

1. Применим к множествам ребер E_1, E_2 операцию сумма по модулю 2 (симметрическая разность): $E = E_1 \oplus E_2 = (E_1 \cup E_2) \setminus (E_1 \cap E_2)$.

Ребра, принадлежащие пересечению E_1, E_2 , будут удалены (будут удалены ребра, составляющие подцепочку, проходящую дважды). Множество E будет состоять из ребер, составляющих простую цепочку, соединяющую две заданные вершины.

2. Упорядочим ребра из E в простую цепочку.

Эта цепочка и будет простой цепочкой, соединяющей две заданные вершины.

Отметим, что простая цепочка, соединяющая две заданные вершины, без учета порядка ребер — минимальный связный подграф, содержащий две заданные вершины. Именно такой подграф получается на шаге 1 правила вычисления простой цепочки.

Приведенные два шага получения простой цепочки из двух смежных цепочек будем рассматривать как одну операцию. Обозначим эту операцию \oplus . Назовем операцию — *композиция по модулю 2*.

Можно сказать, что операция *композиция по модулю 2* выполняется под двумя простыми смежными цепочками и ее результат — простая цепочка, соединяющая несмежные концевые вершины цепочек-операндов. Если у цепочек-операндов смежные обе концевые вершины, то результат операции — пустая цепочка. Пустую цепочку будем обозначать 0.

ОПЕРАЦИЯ КОМПОЗИЦИЯ ПО МОДУЛЮ 2 НА МНОЖЕСТВЕ ПРОСТЫХ ЦЕПОЧЕК И ЕЕ СВОЙСТВА

Обозначим $P = P(E)$ — множество всех простых цепочек, порождаемых множеством ребер графа G . Для смежных цепочек из множества P введем бинарную операцию *композиция по модулю 2*, определенную выше.

Множество P замкнуто относительно операции \oplus , т. е. $\forall p, q \in P$, где p, q — смежные: $p \oplus q \in P$. Эта операция — частично определенная, т. к. применима только к смежным цепочкам. Для обозначения частичной алгебры, определяемой множеством P и операцией \oplus , используется обозначение $\langle P; \oplus \rangle$.

Свойства операции \oplus :

A. Ассоциативность. Следует из определения операции \oplus .

B. Нейтральный элемент — пустая цепочка 0: $\forall p \in P: p \oplus 0 = 0 \oplus p = p$.

C. Обратным элементом для любой цепочки является сама эта цепочка: $\forall p \in P: p \oplus p = 0$.

D. Коммутативность: $\forall p, q \in P$ и p, q — смежные: $p \oplus q = q \oplus p$. Следует из определения операции \oplus .

Пара $\langle P; \oplus \rangle$ — не образует группы, т. к. операция \oplus — частично определенная, но многие групповые свойства выполняются.

Свойство 1. Обратный элемент единственный.

Свойство 2. Для смежных цепочек из P выполняются следующие соотношения:

1. $(p \oplus q)^{-1} = q^{-1} \oplus p^{-1}$.
2. $(p \oplus q) = (p \oplus r) \Rightarrow q = r$.
3. $(q \oplus p) = (r \oplus p) \Rightarrow q = r$.
4. $((p)^{-1})^{-1} = p$.

Свойство 3. В частичной алгебре $\langle P; \oplus \rangle$ уравнение $a \oplus x = b$ однозначно разрешимо тогда и только тогда, когда цепочки a и b — смежные. Решение имеет вид $x = b \oplus a$.

ПОЛНЫЕ ПОДСИСТЕМЫ ПРОСТЫХ ЦЕПОЧЕК

Пусть $M \subset P$ — некоторое подмножество простых цепочек из P .

Определение 2. Подмножество простых цепочек M называется полным (относительно операции \oplus), если любая простая цепочка из P может быть получена из цепочек множества M с использованием операции \oplus .

Пример 1. $M_1 \subset P$ — подмножество всех простых цепочек длины 1 («однореберных» цепочек). M_1 — подмножество полное, т. к. любая простая цепочка — это конечная последовательность смежных ребер и, следовательно, может быть представлена в виде

$$p = p_1 \oplus p_2 \oplus \dots \oplus p_n,$$

где $p = p_1, p_2, \dots, p_n$ — смежные ребра, составляющие эту цепочку.

Пример 2. $M_0 \subset P$ — подмножество всех простых цепочек, соединяющих некоторую выделенную вершину (корень) с каждой из остальных вершин графа. M_0 — подмножество полное, т. к. любую цепочку $p \in M_1$ («однореберную» цепочку) можно представить в виде

$$p = p_1 \oplus p_2.$$

Здесь $p_1, p_2 \in M_0$, а значит, смежные в вершине, являющейся корнем, и такие что у p_1 вторая концевая вершина совпадает с одной из концевых вершин цепочки p , а у p_2 вторая концевая вершина совпадает с другой концевой вершиной цепочки p .

Определение 3. Базис — это минимальная полная подсистема цепочек.

Минимальная полная подсистема — это полное подмножество простых цепочек, никакое собственное подмножество которого не является полным.

Пример 3. $M_1 \subset P$ — подмножество всех простых цепочек длины 1.

M_1 — подмножество полное. При удалении любого ребра подмножество M_1 перестает быть полным. Следовательно, M_1 — базис.

Пример 4. $M_0 \subset P$ подмножество всех простых цепочек, соединяющих некоторую выделенную вершину с каждой из остальных вершин графа.

M_0 подмножество полное. При удалении любой цепочки подмножество M_0 перестает быть полным. Следовательно, M_0 — базис.

Для любой из вершин графа можно построить систему простых цепочек, соединяющих эту вершину с каждой из остальных вершин графа. Каждая из таких систем является базисом.

Определение 4. Канонический базис — это базис, состоящий из простых цепочек, соединяющих заданную вершину графа с каждой из остальных вершин (канонический базис, порождаемый заданной вершиной).

В качестве вершины, относительно которой строится канонический базис, может использоваться любая вершина графа. Поэтому канонических базисов столько, сколько вершин в графе.

ПЕРЕХОД ОТ ОДНОГО КАНОНИЧЕСКОГО БАЗИСА К ДРУГОМУ

Пусть ν_1, ν_2 — вершины графа, B_1 — канонический базис, порождаемый вершиной ν_1 , $\langle \nu_1, \nu_2 \rangle$ — цепочка, соединяющая вершины ν_1, ν_2 . Канонический базис B_2 , порождаемый вершиной ν_2 , получается по следующему правилу:

Для каждой цепочки из B_1 выполняем операцию \oplus с цепочкой $\langle \nu_1, \nu_2 \rangle$. Получившиеся цепочки и составляют канонический базис B_2 , порождаемый вершиной ν_2 .

Для поиска простых цепочек, соединяющих две заданные вершины (обозначим эти вершины ν_1, ν_2), возможно использование следующего алгоритма:

1. Одна из вершин берется в качестве корня. Обозначим эту вершину ν_0 .
2. Строятся простые цепочки, соединяющие корень с каждой из остальных вершин графа (канонический базис). Обозначим это множество цепочек M_0 .
3. Из множества M_0 выбираются две цепочки p_1, p_2 такие, что каждая из них соединяет корень с одной из двух заданных вершин (цепочка $p_1 = \langle \nu_1, \nu_0 \rangle$, цепочка $p_2 = \langle \nu_2, \nu_0 \rangle$).
4. Цепочка p , соединяющая две заданные вершины ν_1, ν_2 , получается как результат применения операции \oplus к цепочкам p_1, p_2 .

ЗАМЕЧАНИЯ К АЛГОРИТМУ ПОСТРОЕНИЯ ПРОСТЫХ ЦЕПОЧЕК

1. Алгоритм имеет смысл применять в том случае, если для данного графа придется многократно решать задачу нахождения простой цепочки, соединяющей две несовпадающие вершины. В этом случае, построение канонического базиса M_0 выполняется один раз, и для построения любой цепочки используются цепочки из M_0 .

2. Конкретные процедуры реализации пунктов алгоритма не фиксируются. Временная сложность алгоритма будет зависеть от выбора этих процедур. Существуют методы поиска в ширину, позволяющие построить сразу все цепочки множества M_0 . Сложность таких методов не зависит от того, строим мы одну или все цепочки, соединяющие с корнем остальные вершины.

2. ЗАДАЧА ШТЕЙНЕРА ДЛЯ СВЯЗНОГО НЕОРИЕНТИРОВАННОГО АЦИКЛИЧЕСКОГО ГРАФА

ОПЕРАЦИЯ «СОЕДИНЕНИЕ ПО ШТЕЙНЕРУ» И ЕЕ СВОЙСТВА

Пусть $G = \langle V_G, E_G \rangle$ — конечный связный неориентированный ациклический граф. $V_G = \{\nu_1, \nu_2, \dots, \nu_n\}$ — множество вершин, $E_G = \{e_1, e_2, \dots, e_m\}$ — множество ребер графа G .

Обозначим $S = S(G)$ — множество всех связных подграфов, порождаемых множеством элементов графа G . Подграф, не содержащий ни одной вершины (пустой подграф) считаем связным по определению и обозначаем 0 . Подграф, состоящий из одной изолированной вершины, также считаем связным.

Пусть $A = (V_A, E_A)$, $B = (V_B, E_B)$ — связные подграфы из S , $p = (V_p, E_p)$ — минимальная из цепочек, соединяющих вершины подграфов A и B .

Так как граф G — связный, то такая цепочка *существует*.

Так как граф G — ациклический, то такая цепочка *единственная*.

Если $A \cap B \neq \emptyset$ (u A и B есть общие элементы), то минимальная цепочка $p = 0$.

Соединим подграфы A и B минимальной цепочкой p . В результате получится связный подграф из S . Используемую операцию соединения назовем *соединение по Штейнеру*. Для обозначения этой операции используем символ $\$$. Тогда подграф, получаемый подграфов A и B применением соединения по Штейнеру, определяется как

$$C = A\$B = A \cup B \cup p,$$

где $V_C = V_A \cup V_B \cup V_p$ — множество вершин графа C , $E_C = E_A \cup E_B \cup E_p$ — множество ребер графа C .

Таким образом, операция *соединение по Штейнеру* выполняется над двумя связными подграфами и ее результат — связный подграф, т. е. множество S замкнуто относительно операции $\$$.

Для обозначения алгебры, определяемой множеством S и операцией $\$$, используется обозначение $\langle S; \$ \rangle$.

Свойства операции $\$$:

A. Ассоциативность. Следует из определения операции $\$$.

B. Нейтральный элемент — пустой подграф 0 : $\forall A \in S: A\$0 = 0\$A = A$.

C. Ассоциативность. Следует из определения операции $\$$.

Следовательно, пара $S; \$$ — образует коммутативную полугруппу с нейтральным элементом.

D. В полугруппе $\langle S; \$ \rangle$ уравнение $A\$X = B$ разрешимо тогда и только тогда, когда подграф A содержится в подграфе B и $B \setminus A$ — связный подграф. Решение может быть не единственным.

Пусть $M \subset S$ некоторое подмножество связных подграфов из S .

Определение 5. Подмножество связных подграфов M называется полным (относительно операции $\$$), если любой связный подграф из S может быть получен из подграфов множества M с использованием операции $\$$.

Пример 5. $M_1 \subset S$ — подмножество всех одновершинных подграфов и всех ребер графа G . M_1 — подмножество полное, т. к. любой связный подграф из S может быть получен последовательным соединением его вершин минимальными цепочками, составленными из ребер графа G и может быть представлен в виде

$$B = \{\nu_1\}\$\{\nu_2\}\$\dots\$\{\nu_k\},$$

где $\nu_1, \nu_2, \dots, \nu_k$ — вершины подграфа.

При выполнении операции $\$$ необходимо находить минимальную цепочку, соединяющую подграфы — операнды. Рассмотрим возникающие здесь ситуации.

Пусть $A = (V_A, E_A)$, $B = (V_B, E_B)$ — связные подграфы из S , $p = (V_p, E_p)$ — минимальная из цепочек, соединяющих вершины подграфов A и B .

1. Если $A \cap B \neq \emptyset$ (у A и B есть общие элементы), то минимальная цепочка $p = 0$.
2. Если A и B — одновершинные подграфы, то минимальная цепочка p — это цепочка, соединяющая две заданные вершины. Подграф $A\$B$ это сама минимальная цепочка p .
3. Если $A \cap B = \emptyset$ (у A и B нет общих элементов), то минимальную цепочку p необходимо найти среди цепочек, соединяющих вершины из A и B .

Если в ациклическом графе одна из вершин выбрана в качестве корня, то простая цепочка, соединяющая две вершины графа, проходит через корень минимального из поддеревьев, содержащего обе соединяемые вершины. Любой связный подграф содержится в минимальном из поддеревьев, содержащем все вершины этого подграфа, и содержит вершину, являющуюся корнем этого минимального поддерева. Если у подграфов A и B нет общих элементов, то минимальная цепочка, соединяющая вершины из A и B — это цепочка между корнями минимальных поддеревьев, содержащих подграфы A и B . Следовательно, при выполнении соединения по Штейнеру необходимо определить цепочку, соединяющую корни минимальных поддеревьев, содержащих подграфы A и B .

Подход к поиску минимальной цепочки, соединяющей подграфы A и B , может быть основан на операции композиции по модулю 2, определенной ранее.

Пусть M_0 — подмножество всех простых цепочек, соединяющих некоторую выделенную вершину (корень) с каждой из остальных вершин дерева. В силу полноты M_0 относительно операции \oplus , любую цепочку p можно представить в виде

$$p = p_1 \oplus p_2,$$

где p_1, p_2 подходящие цепочки из M_0 . Если нам известны корни минимальных поддеревьев, содержащих подграфы A и B , то для поиска минимальной цепочки воспользуемся операцией \oplus .

РЕШЕНИЕ ЗАДАЧИ ШТЕЙНЕРА ДЛЯ АЦИКЛИЧЕСКОГО ГРАФА

Если каждую из вершин подмножества U рассматривать как одновершинный подграф, то искомым подграф T получается в результате применения соединения по Штейнеру к вершинам из U , т. е., если $U = \{u_1, u_2, \dots, u_k\} \subset V$ — заданное подмножество вершин, $\{u_1\}, \{u_2\}, \dots, \{u_k\}$ — соответствующие одновершинные подграфы, то

$$T = \{u_1\} \oplus \{u_2\} \oplus \dots \oplus \{u_k\}.$$

Пусть M_0 — подмножество всех простых цепочек, соединяющую некоторую выделенную вершину (корень) с каждой из остальных вершин дерева. Для поиска минимальных цепочек, соединяющих вершины из U , воспользуемся операцией \oplus . Пусть $\{p_1, p_2, \dots, p_k\}$ — цепочки, соединяющие соответствующие вершины из U с корнем, $\{E_1, E_2, \dots, E_k\}$ — множества ребер этих цепочек. Тогда решение задачи Штейнера запишется

Теорема 1.

$$T = \{u_1\} \oplus \{u_2\} \oplus \dots \oplus \{u_k\} = p_1 \oplus p_2 \oplus \dots \oplus p_k = \left(\bigcup_{i=1}^k E_i \right) \setminus \left(\bigcap_{i=1}^k E_i \right).$$

Доказательство. При доказательстве используются следующие равенства и обозначения:

$$\forall A, B, C : (A \setminus C) \cup (B \setminus C) = (A \cup B) \setminus C, \quad (**)$$

$$C \subset B \subset A \Rightarrow (A \setminus B) \cup (B \setminus C) = (A \setminus C). \quad (*)$$

Обозначим $T_{ij\dots k}$ — построенную часть дерева Штейнера, содержащую заданные вершины u_1, u_2, \dots, u_k .

Для сокращения записи знак операции пересечения будем опускать, так что $A \cap B = AB$.

$$T_{12} = T_1 \oplus T_2 = E_1 \oplus E_2 = (E_1 \cup E_2) \setminus E_1 E_2$$

$$\begin{aligned}
T_{123} &= T_{12} \$ T_3 = T_{12} \cup [(E_1 E_2 \cup E_3) \setminus E_1 E_2 E_3] = ((E_1 \cup E_2) \setminus E_1 E_2) \cup ((E_1 E_2 \cup E_3) \setminus E_1 E_2 E_3) = \\
&= ((E_1 \cup E_2) \setminus E_1 E_2) \cup (E_1 E_2 \setminus E_1 E_2 E_3) \cup (E_3 \setminus E_1 E_2 E_3) = \\
&= |E_1 E_2 E_3 \subset E_1 E_2 \subset E_1 \cup E_2, (*)| = ((E_1 \cup E_2) \setminus E_1 E_2) \cup (E_1 E_2 \setminus E_1 E_2 E_3) = (E_1 \cup E_2) \setminus E_1 E_2 E_3 | = \\
&= ((E_1 \cup E_2) \setminus E_1 E_2 E_3) \cup (E_3 \setminus E_1 E_2 E_3) = |(**)| = (E_1 \cup E_2 \cup E_3) \setminus E_1 E_2 E_3
\end{aligned}$$

...

$$\begin{aligned}
T_{12\dots k-1} &= \dots = (E_1 \cup E_2 \cup \dots \cup E_{k-1}) \setminus E_1 E_2 \dots E_{k-1} \\
T_{12\dots k} &= T_{12\dots k-1} \$ T_k = T_{12\dots k-1} \cup [(E_1 E_2 \dots E_{k-1} \cup E_k) \setminus E_1 E_2 \dots E_k] = \\
&= ((E_1 \cup E_2 \cup \dots \cup E_{k-1}) \setminus E_1 E_2 \dots E_{k-1}) \cup ((E_1 E_2 \dots E_{k-1} \cup E_k) \setminus E_1 E_2 \dots E_k) \cup (E_k \setminus E_1 E_2 \dots E_k) = \\
&= |E_1 E_2 \dots E_k \subset E_1 E_2 \dots E_{k-1} \subset E_1 \cup E_2 \cup \dots \cup E_{k-1} (*)| = \\
&= ((E_1 \cup E_2 \cup \dots \cup E_{k-1}) \setminus E_1 E_2 \dots E_{k-1}) \cup (E_k \setminus E_1 E_2 \dots E_k) = |(**)| = \\
&= (E_1 \cup E_2 \cup \dots \cup E_k) \setminus E_1 E_2 \dots E_k = \left(\bigcup_{i=1}^k E_i \right) \setminus \left(\bigcap_{i=1}^k E_i \right)
\end{aligned}$$

ПОЛИНОМИАЛЬНАЯ РАЗРЕШИМОСТЬ ЗАДАЧИ ШТЕЙНЕРА ДЛЯ АЦИКЛИЧЕСКОГО ГРАФА

Рассматриваемый алгоритм решения задачи построения дерева Штейнера состоит из двух основных шагов:

1. Построение подмножество всех простых цепочек, соединяющую некоторую вершину (корень) с каждой из остальных вершин дерева.
2. Применение теоремы 1 к цепочкам, соединяющим корень с вершинами из множества U .

Общая сложность этих двух последовательных шагов

$$t(n, m, k) = t_1(\cdot) + t_2(\cdot),$$

где $t(n, m, k)$ — общая сложность алгоритма, n — число вершин, $m = n - 1$ — число ребер исходного графа, k — число вершин в множестве U , $t_1(\cdot), t_2(\cdot)$ — сложность первого и второго шагов соответственно.

Шаг 1 может быть реализован любым известным алгоритмом, например, алгоритм Дейкстры. Сложность этого алгоритма для n цепочек $O(n^3)$.

На шаге 2 из списка, состоящего из n цепочек, выбираются k цепочек. Сложность этого выбора $O(mk)$.

Так как $k \leq n$, то общая оценка сложности $t(n, m, k) = O(n^3) + O(mn)$ — полиномиальна от длины начальной информации.

СПИСОК ЛИТЕРАТУРЫ

1. Емеличев В.А., Мельников О.И., Сарванов В.И., Тышкевич Р.И. Лекции по теории графов. — М.:Наука, 1990. — 384с.
2. Евстигнеев В.А. Применение теории графов в программировании. — М.:Наука, 1985. — 352с.