Остовные деревья

Остовные деревья

Определение. Остовным деревом связного неориентированного графа $\Gamma = (X, U, \Phi)$ называется дерево $\Gamma_0 = (X, U_0, \Phi)$, являющееся подграфом графа Γ и содержащее все его вершины.

Утверждение. Дерево с n вершинами содержит $n - 1$ ребро.

Доказательство. Доказательство проведём по индукции числа вершин. Заметим, что в дереве найдётся вершина, степень которой равна единице (висячая вершина). Действительно, в противном случае, если степени вершины ≥ 2, можно построить цикл, что противоречило бы определению дерева. Цикл строится следующим образом. Выполним проход по рёбрам графа, начиная с произвольной вершины. Так как степени ≥ 2, то, попав в вершину в первый раз, всегда можно выйти из неё. Исходный граф конечный и связный. Следовательно, наступит момент, когда мы вновь попадём в уже пройденную вершину. Т.е. цикл существует.

Условие индукции для $n = 2$ выполняется, дерево с двумя вершинами содержит одно ребро. Предположим теперь, что утверждение выполняется для деревьев, число вершин у которых меньше n. Рассмотрим дерево с n вершинами. Удалим из этого дерева висячую вершину и инцидентное ей ребро. Очевидно, что оставшийся граф будет связным и без циклов, т.е. будет с деревом с $n - 1$ вершиной. Тогда по предположению индукции оставшаяся часть графа содержит $n - 2$ ребра, а значит, исходное дерево должно иметь их $n - 1$.

Практическую значимость остовых деревьев даёт популярная форма задачи Кэли. Необходимо соединить n городов железнодорожными линиями так, чтобы не строить лишних дорог. Известна стоимость строительства дорог для каждой пары городов. Какова должна быть сеть дорог, соединяющая все города и имеющая минимальную возможную стоимость? Аналогичные вопросы возникают при проектировании линий электропередач и д.р.

В терминах теории графов задачу можно сформулировать следующим образом. Рассмотрим граф $\Gamma = (X, U, \Phi)$, где X – города, U – дороги. Каждому ребру $u \in U$ назначим вес $w(u)$ - стоимость строительства дороги u. Задача состоит в том, чтобы построить связный граф $\Gamma_0 = (X, U_0, \Phi)$, содержащий все вершины, с минимальным весом $W(\Gamma_0) = \sum_{u \in U_0} w(u)$. Очевидно, что Γ_0 - дерево, в противном случае можно было бы удалить одно ребро, не нарушая связности Γ_0 и уменьшая сумму весов его рёбер.

Определение. Минимальным остовным деревом (лесом) называется остовное дерево (лес) с минимальным общим весом его рёбер.

Жадный алгоритм построения минимального остовного дерева

Минимум остовных деревьев графа $\Gamma = (X, U, \Phi)$ можно найти, применяя процедуру исследования рёбер в порядке возрастания их весов. Другими словами, на каждом шаге выбирается новое ребро с наименьшим весом, не образующее циклов с уже выбранными рёбрами. Процесс продолжается до тех пор, пока не будет выбрано $|X| - 1$ ребро. Рассмотренная процедура называется жадным алгоритмом.

Реализация данной схемы может быть выполнена следующим образом. Для каждой вершины $X = \{x_1, x_2, \dots, x_n\}$ графа $\Gamma = (X, U, \Phi)$ формируются начальные тривиальные компоненты связности $T_i = (X_i, U_i, \Phi)$, где $X_i = \{x_i\}, U_i = \emptyset, X_i \cap X_j = \emptyset, i \neq j, X = \bigcup_{i=1}^{|X|} X_i, i = 1,2, \dots, |X|$. Компоненты T_i являются деревьями, объединение $T = \bigcup_i T_i$ которых даёт начальное приближение строящегося остовного дерева $\Gamma_0 = (X, U_0, \Phi)$.

Включение в строящееся остовное дерево Γ_0 выбранного ребра на очередном шаге жадного алгоритма выполняется слиянием $T_i = T_i \cup T_j$ $(X_i = X_i \cup X_j, U_i = U_i \cup U_j)$ двух компонент T_i и T_j, которым принадлежит по вершине нового ребра, и включением самого ребра в объединённое множество $U_i = U_i \cup U_j$ рёбер. Процесс роста объединения $T = \bigcup_i T_i$ компонент к остовному дереву $\Gamma_0 = (X, U_0, \Phi)$ продолжаем до тех пор, пока не будет включено $|X| - 1$ ребро. Справедливость жадного алгоритма является следствием следующих двух лемм.

Лемма 1. Пусть $\Gamma = (X, U, \Phi)$ - связанный неориентированный граф и $\Gamma_0 = (X, U_0, \Phi)$ - произвольное остовное дерево для него. Тогда верны следующие утверждения:

1. $\forall x_1, x_2 \in X$ существует единственная между ними цепь в Γ_0
2. Если к Γ_0 добавить ребро из $U \backslash U_0$, то возникает ровно один цикл.

Доказательство. Утверждение 1 верно, т.к. в противном случае в Γ_0 существовал бы цикл, что противоречит дереву Γ_0. Утверждение 2 верно, поскольку между вершинами добавляемого ребра уже есть одна цепь, а значит возникает один цикл.

Лемма 2. Пусть $\Gamma = (X, U, \Phi)$ - связанный неориентированный граф, для каждого ребра $u \in U$ определён вес $w(u)$, и $T_i = (X_i, U_i, \Phi)$ - компоненты связности жадного алгоритма, объединение которых $T = \bigcup_i T_i$, согласно алгоритму, растёт к остовному дереву $\Gamma_0 = (X, U_0, \Phi)$, где $i = 1,2, \dots, k$ и $k > 1$. Пусть следующим найденным для включения ребром является ребро $\varepsilon = (x, y)$ наименьшего веса из оставшихся $U \backslash \bigcup_i U_i$ и пусть $x \in X_1, y \notin X_1$. Тогда найдётся остовное дерево Γ_0 для $\Gamma = (X, U, \Phi)$, содержащее рёбра $\bigcup_i U_i \cup \{\varepsilon\}$, вес которого не больше любого другого остовного дерева, содержащего рёбра $\bigcup_i U_i$.

Доказательство. Допустим противное. Пусть существует остовное дерево $\Gamma_0' = (X, \mathrm{U}_0', \Phi)$ для Γ, содержащее $\bigcup_i U_i$ и не содержащее рёбра ε, вес которого меньше веса любого остовного дерева для Γ, содержащего $\bigcup_i U_i \cup \{\varepsilon\}$. По утверждению леммы 1, при добавлении ε к Γ_0' образуется цикл. Этот цикл должен содержать такое ребро $\varepsilon' = (x', y')$, отличное от ε, что $x' \in X_1, y' \notin X_1$, т.к. цикл входит в U_1 по ребру (x, y) и $y \notin X_1$, то он должен и выйти из U_1. Из условия следует, что вес $w(\varepsilon) \leq w(\varepsilon')$, так как $\bigcup_i U_i \subseteq U_0$ и $\bigcup_i U_i \subseteq U_0'$ и ребро ε выбиралось с минимальным весом из оставшихся рёбер $U \backslash \bigcup_i U_i$ без образования циклов, в противном же случае выбор должен был бы пасть на ε', т.к. оно тоже не образует циклов с $\bigcup_i U_i$.

Рассмотрим граф $\Gamma_0 = (X, U_0, \Phi)$, образованный добавлением ε к Γ_0' и удалением ε' из Γ_0'. В Γ_0 нет циклов, так как единственный цикл разорван удалением ребра ε'. Γ_0 остался связным, так как осталась цепь между x' и y'. Таким образом, Γ_0 – остовное дерево. Учитывая, что $w(\varepsilon) \leq w(\varepsilon')$, то вес дерева Γ_0 не больше веса Γ_0'. Остовное дерево Γ_0 содержит $\bigcup_i U_i$ и ε, а это противоречит предположению минимальности Γ_0', что доказывает справедливость жадного алгоритма.

Реализация жадной схемы формирования остовного дерева представлена ниже. Используемые обозначения соответствуют тем обозначениям, которые вводились при обосновании жадной схемы. Вектор $Mark[x]$ меток вершин $x \in X$ графа поддерживает их принадлежность

компонентам связности U_i, из которых формируется рёберный список остовного дерева. Начальные величины $Mark[x_i]$ устанавливаются равными порядковым номерам соттветствующих вершин $x_i \in X$. Далее значения $Mark[x_i]$ корректируются по мере слияния компонент U_i, сохраняя соответствие принадлежности им вершин. Исходный граф задаётся рёберным списком U, выходное остовное дерево также формируется рёберным списком U_0.

Листинг 1

```
for x_i ∈ X do Mark[x_i]:=i;
  Sort(U);
U_0:=∅;
while |U_0| < |X| − 1 do
begin
  (x, y) ∈ U;
  if Mark[x] ≠ Mark[y] then
  begin
    U_0 = U_0 ∪ {(x, y)};
    z:=Mark[y];
    for v ∈ X do
      if Mark[v] = z then
        Mark[v]:=Mark[x];
  end;
  U:=U\{(x, y)};
end;
```

Сложность жадного алгоритма

Жадный алгоритм требует предварительной сортировки рёбер по весу. Изученные нами методы позволяют сортировать со сложностью $O(|U| * \log(|U|))$. Помимо сортировки, в алгоритме выполняется ещё одна сложная процедура слияния подмножеств U_x и U_y. Сложность данной операции при полном переборе вершин данных подмножеств составляет $O(|X|^2)$. Т.о. полная сложность алгоритма составляет $O(|X|^2 + |U| * \log(|U|)) = O(|X|^2)$.

Алгоритм ближайшего соседа построения остовного дерева

Данный метод построения минимального остовного дерева не требует ни сортировки, ни проверки на цикличность на каждом шаге.

1. Построение остовного дерева Γ_0 начинается с произвольной вершины x_1.
2. Затем, среди рёбер, инцидентных x_1, выбираем ребро (x_1, x_2) с наименьшим весом и включаем его в дерево Γ_0.
3. Повторяя процесс, выполняем поиск наименьшего о весу ребра, соединяющего вершины x_1 и x_2 с некоторой другой вершиной графа x_3.
4. Процесс включения рёбер продолжаем до тех пор, пока все вершины исходного графа Γ не будут включены в дерево Γ_0. Построенное дерево будет минимальным остовным.

Доказательство. Аналогично доказательству для жадного алгоритма. Реализация схемы ближайшего соседа описана ниже. Исходный граф $\Gamma = (X, U, \Phi)$ представлен матрицей весов $W_e = [w_{ij}]$, веса отсутствующих рёбер полагаются равными $+\infty$. Остовное дерево $\Gamma_0 = (X_0, U_0, \Phi)$ формируется посредством рёберного списка U_0 и списка вершин X_0. В качестве меток вершин устанавливаются их порядковые номера $X = \{1,2, \dots, |X|\}$. Для каждой вершины $x \in X$ графа, ещё не включённой в остовное дерево, поддерживается минимальное расстояние до множества ранее включённых вершин в X_0. Это осуществляется с помощью двух векторов $dist[x]$ и $prev[x]$,

где $dist[x]$ равно минимальному расстоянию от $x \in X$ до вершины $prev[x] \in X_0$. Обновление значений векторов $dist[x]$ и $prev[x]$ выполняется на каждом шаге алгоритма при пополнении X_0 новой вершиной.

Листинг 2

```
X:={1,2, … , |X|};
nX:= |X|;
v:=rand(1, |X|);
X0:={v};
X:=X\{v};
U0:=∅;
Wt:=0;
for x ∈ X do
begin
   dist[x] ≔ We[x, v];
   prev[x]:=v;
end;
while |X0| ≠ nX do
begin
   dist[v]:=min dist[x];
            x∈X
   X0:=X0 ∪ {v};
   X:=X\{v};
   U0:=U0 ∪ {prev[v], v};
   Wt:=Wt + dist[v];
   for x ∈ X do
     if dist[x] > We[v, x] then
     begin
        dist[x]:=We[v, x];
        prev[x]:=v;
     end;
end;
```

Сложность алгоритма ближайшего соседа

Сложность алгоритма определяется двумя вложенными циклами по числу вершин. В каждом из циклов выполняется константное число операций. Следовательно, сложность его составляет $O(|X^2|)$.