

Методы дискретной оптимизации

Остовное дерево (каркас). Алгоритмы
построения остовного дерева

Терминология, известные факты

Определение. Остов (каркас) связного графа - дерево, содержащее все вершины графа. Определяется неоднозначно.

Определение. Цикломатическое (или циклический ранг) число графа: $\nu(G) = m - n + c$, где n – число вершин, m – число ребер, c – число компонент связности графа.

Определение. Коциклический ранг (или коранг) графа: $\nu^*(G) = n - c$

Определение. Ребра графа, не входящие в остов, называются хордами.

Определение. Цикл, получающийся при добавлении к остову графа его хорды, называется фундаментальным относительно этой хорды.

Свойства:

- 1 Число ребер неориентированного графа, которые необходимо удалить для получения остова, не зависит от последовательности их удаления и равно цикломатическому числу графа.
- 2 Неориентированный граф G является лесом тогда и только тогда, когда $\nu(G) = 0$.
- 3 Неориентированный граф G имеет единственный цикл тогда и только тогда, когда $\nu(G) = 1$.
- 4 Остов неориентированного графа имеет $\nu^*(G)$ ребер.

Свойства

Пусть $G_1 = (V, E_1)$ и $G_2 = (V, E_2)$ – два остовных дерева графа G .

Расстоянием между ними называется величина $d(G_1, G_2) = |E_1 \setminus E_2|$.

Множества вершин в деревьях можно поменять ролями, т.к.

$$|E_1| = |E_2| = n - 1.$$

Если $d(G_1, G_2) = 1$, то $(E_1 \cup E_2) \setminus (E_1 \cap E_2) = \{(e_1, e_2)\}$, $e_i \in E_i, i = 1, 2$, тогда одно дерево получается из другого заменой одной дуги. Такой переход называется элементарным переходом от дерева к соседнему.

Теорема 1

Если для двух деревьев $G_0, G_k : d(G_0, G_k) = k$, то второе получается из первого (и наоборот) за k элементарных переходов.

Доказательство. Пусть $\{e_i^0 \in E_0 \setminus E_k, e_i^k \in E_k \setminus E_0, i = 1, 2, \dots, k\}$. Добавим в множество ребер E^0 ребро e_1^k , тогда в G_0 возникнет цикл, в котором существует ребро $e \notin E_k$, т.к. G_k – дерево. Действительно, если в этом цикле присутствуют ребра исключительно из E_k , то G_k не является деревом. Удалим из цикла это ребро, тогда цикл исчезнет. Таким образом из G_0 будет удалено ребро множества E_0 и вместо него помещено ребро из E_k , при этом расстояние между деревьями G_k и $G_0^1 = (V, (E_0 \setminus e) \cup e_1^k)$ станет равным $k - 1$. Продолжая процесс далее, получим дерево $G_k^k = G_k$.

Взвешенные графы. Задача Штейнера

Определение. Пусть $G = (V, E)$ – связный простой граф. G называется взвешенным графом, если задана функция $c(e)$, называемая весом (длиной) ребра. Обозначение: $G = (V, E, c)$.

Для произвольной части H графа G ее весом называется сумма весов входящих в нее ребер. Обозначение: $c(H)$.

Пусть H – остовное дерево (каркас) графа, тогда $H = (V, ET)$ и задача поиска остовного дерева минимального веса формулируется как

$$c(T) = \sum_{e \in ET} c(e) \rightarrow \min$$

В теории графов широко известна задача Штейнера: на плоскости заданы несколько точек; нужно соединить их отрезками прямых таким образом, чтобы суммарная длина отрезков была наименьшей. Главное отличие от задачи о минимальном остовном дереве при этом заключается в том, что разрешается добавлять дополнительные точки ветвления с целью ещё сильнее уменьшить сумму длин рёбер. Задача о дереве Штейнера является NP-полной.

Разрезы графа

Операция вырезания-вставки ребра в остов является основным элементом построения каркаса минимального веса.

Определение. Разбиение $(S, V \setminus S)$ множества вершин называется разрезом графа G .

Определение. Ребро (u, v) пересекает разрез $(S, V \setminus S)$, или является перешейком данного разреза, если концы этого ребра находятся в разных частях разреза.

Определение. Ребро (u, v) называется легким ребром для разреза, если это ребро имеет минимальный вес среди всех перешейков данного разреза.

Определение. Множество ребер *согласовано с разрезом* $(S, V \setminus S)$, если ни одно из ребер A не является перешейком для этого разреза. Если ребра из A соединяют хотя бы одну вершину из S , то все вершины, инцидентные ребрам из A , лежат в S . Поэтому можно сокращенно обозначать $A \subseteq S$ факт согласованности A с разрезом $(S, V \setminus S)$.

Теорема о минимальном разрезе

Теорема 2

Пусть $(S, V \setminus S)$ – разрез графа G и множество ребер T в E образует минимальный каркас G . При выполнении условий:

1. Множество ребер $A \subseteq S$ согласовано с разрезом,
2. Легкое для данного разреза ребро $(u, v) \notin T$.

существует минимальный каркас $T' : (A \cup (u, v)) \subset T'$.

Доказательство. Считаем, что ребро $(u, v) \notin T$, в противном случае $T' = T$. В каркасе T существует ребро (x, y) – перешеек разреза, поскольку T – остов G . При этом по условию $A \subseteq S \Rightarrow (x, y) \notin A$. Положим $T' = (T \setminus (x, y)) \cup (u, v)$. Тогда вес данного каркаса не превосходит вес T , поскольку (u, v) – легкое ребро для разреза $(S, V \setminus S)$, что означает $c(u, v) \leq c(x, y) \Rightarrow c(T') \leq c(T)$. Теорема доказана.

Задача

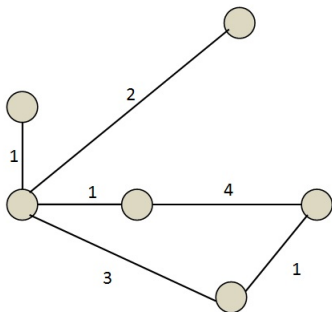
Пусть $(S, V \setminus S)$ – разрез графа G , множество ребер T в E образует минимальный каркас G , множество ребер $A \subseteq S \cap T$ и существует минимальный каркас $T' : (A \cup (u, \nu)) \subset T'$. Следует ли отсюда, что (u, ν) – легкое ребро для данного разреза? Привести контрпример.

Решение. Рассмотрим граф, задаваемый матрицей расстояний: бесконечность означает отсутствие ребра, 0 означает расстояние от вершины до нее самой.

$$A = \begin{pmatrix} 0 & 1 & \infty & 3 & 2 & 1 \\ 1 & 0 & 4 & \infty & \infty & \infty \\ \infty & 4 & 0 & 1 & \infty & \infty \\ 3 & \infty & 1 & 0 & \infty & \infty \\ 2 & \infty & \infty & \infty & 0 & \infty \\ 1 & \infty & \infty & \infty & \infty & 0 \end{pmatrix}$$

Зададим $S = \{1, 2\}$, $A = (1, 2)$. Минимальный остов состоит из ребер $\{(1, 2), (3, 4), (1, 4), (1, 5), (1, 6)\}$. При этом $(1, 4)$ – перешеек длины 3 – часть минимального остова, но минимальный перешеек $(1, 6)$ имеет длину 1, перешеек $(1, 5)$ длины 2.

Пример



Следствие. Пусть $G_A = (V, A)$ – лес для связного неориентированного графа $G = (V, E)$, $C = (V_C, E_C)$ – компонента леса, – подмножество, входящее в минимальный каркас T для G . Если (u, v) – легкое ребро, соединяющее C с другим компонентом леса, то существует минимальный каркас T_1 , содержащий $A \cup (u, v)$. В принятой терминологии ребро (u, v) – безопасное ребро для .

Алгоритм Краскала (1956)

1. Пусть O_n – граф с n вершинами и пустым количеством ребер. Строим граф $T_1 = e_1$, e_1 – ребро минимального веса в E . Это ребро определяется неоднозначно и решение самой задачи не единственно.
2. Пусть построен граф $T_i, i < n - i$. Строим граф $T_{i+1} = T_i + e_{i+1}$, где ребро e_{i+1} является ребром минимального веса среди ребер, не входящих в T_i и не составляющих циклов с ребрами из T_i . Это означает: алгоритм находит безопасное ребро для добавления в растущий лес с помощью поиска ребра (u, v) с минимальным весом среди всех ребер, соединяющих 2 дерева в лесу.

Алгоритм Краскала

Теорема 3

При $i < n - 1$ граф $T_{i+1} = T_i + e_{i+1}$ определен корректно, а T_{n-1} – остовное дерево минимального веса.

Доказательство.

1. Граф T_i имеет $i < n - 1$ ребер и не может быть связным. Значит, в графе G имеется ребро, обозначим его e_{i+1} , не составляющее циклов с ребрами графа T_i . Следовательно, определение $T_{i+1} = T_i + e_{i+1}$ корректно.
2. Рассмотрим T_{n-1} , по построению это дерево, надо показать его минимальность. Если предположить, что это не так, то: среди всех остовных деревьев минимального веса графа G выберем остов T , который имеет с деревом T_{n-1} *максимальное число общих ребер*. Рассмотрим номер ребра $i = \min\{k | e_k \in T_{n-1} \setminus T\}$, пусть $e_i = (a, b)$.
3. Рассмотрим в дереве T цепь, соединяющую вершины a, b . Добавим к этому пути ребро $e_i = (a, b)$, получим цикл, в котором есть ребро $e \in T \setminus T_{n-1}$. Такое ребро существует, поскольку в противном случае T_{n-1} содержит цикл.

Алгоритм Краскала

4. Заменяем в дереве T ребро e на e_i , получим новый остов $T' = T - e + e_i$. По предположению остов T – остов минимального веса, откуда $c(T') = c(T) + c(e_i) - c(e) \geq c(T) \Rightarrow c(e_i) \geq c(e)$.
5. С другой стороны, если проследить процесс построения дерева T_{n-1} , то нетрудно заметить, что после выбора ребер $e_1, e_2, \dots, e_{i-1}$ алгоритм Краскала в случае $c(e_i) > c(e)$ выбрал бы не $e_i = (a, b)$, а e , поскольку такой выбор не привел бы к появлению цикла и вес остова был бы меньше. Отсюда следует, что $c(T') = c(T)$, т.е. T' – также остов минимального веса.
6. При этом $|T' \cap T_{n-1}| > |T \cap T_{n-1}|$, что противоречит выбору T . Полученное противоречие доказывает теорему.

Алгоритм Прима-Дейкстры

Данный алгоритм отличается от предыдущего только тем, что не просто ациклический граф, но дерево строится на каждом шаге. Укажем шаги алгоритма.

1. Выбирается ребро $e_1 = (a, b)$ минимального веса и строится дерево $T_1 : V_1 = \{a, b\}, E_1 = \{e_1\}$.
2. Если дерево T_i уже построено и $i < n - 1$, то среди ребер, соединяющих вершины этого дерева с вершинами графа G , не входящими в T_i , выбираем ребро e_{i+1} минимального веса. Полагаем

$$T_{i+1} : V_{i+1} = \{V_i \cup v : e_{i+1} = (v, c), v \notin V_i, c \in V_i\}, E_{i+1} = \{E_i \cup \{e_{i+1}\}\}$$

Доказательство корректности данного алгоритма аналогично предыдущему случаю. Если число ребер графа G равно n , то число операций пропорционально

$$(n - 1) + (n - 2) + \dots + 2 + 1 = \frac{n(n - 1)}{2}.$$

Таким образом, алгоритм Прима-Дейкстры применим к связному взвешенному графу и имеет сложность $O(n^2)$.