

Распределенные алгоритмы и системы

mk.cs.msu.ru → Лекционные курсы → Распределенные алгоритмы и системы

Блок 16

Основные допущения о весах в задаче маршрутизации
Маршрутизация и свойства графов

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

Напоминание о графах

Неориентированный конечный **граф** — это пара (V, E) , где V — конечное множество **вершин** и E — множество **рёбер** (неупорядоченных пар вершин) без **петель** (рёбер вида (v, v) , $v \in V$)

Вершина v и ребро e **инцидентны**, если $\exists w : e = (v, w)$

Вершины v и w **смежны** (соединены ребром; $v - w$) в графе (V, E) , если $(v, w) \in E$

Конечный **путь** в графе (из v_1 в v_k) — это конечная последовательность вершин, в которой каждые две соседние вершины соединены ребром:

$$v_1 - v_2 - \dots - v_k$$

Напоминание о графах

Вершина w **достижима** из вершины v , если существует путь из v в w

Граф **связен**, если в нём любая вершина достижима из любой другой

Простой путь — это путь, все вершины которого попарно различны

Цикл в графе — это путь, в котором

- ▶ все рёбра попарно различны и
- ▶ первая и последняя вершины совпадают

Дерево — это связный граф без циклов

Другое определение дерева: это связный граф, в котором количество рёбер на 1 меньше количества вершин

Остовное дерево графа (V, E) — это дерево (V, E') , где $E' \subseteq E$

Допущения

Независимо от конкретного определения стоимости пути, при обсуждении задачи маршрутизации будет использоваться несколько допущений (соглашений)

Пусть $C(P)$ — вес (стоимость) пути P

Тогда:

1. Значение $C(P)$ зависит только от P и определяется однозначно
В частности, вес маршрута (для заданного пакета) не зависит от устройства других маршрутов (для других пакетов)
2. Вес **последовательного сцепления** путей равен сумме весов этих путей:
$$C(v_1 - \dots - v_k - \dots - v_m) = C(v_1 - \dots - v_k) + C(v_k - \dots - v_m)$$

В частности **пустой путь** (содержащий одну вершину и ни одного ребра) имеет вес 0
3. В графе топологии нет циклов отрицательного веса

Будем называть этот набор ограничений **основными допущениями**

Свойства графов

Путь в графе из заданной вершины v в заданную вершину w будем называть **оптимальным** для заданной функции стоимости, если не существует путей из v в w меньшего веса

Лемма (о простых путях). В основных допущениях для любых вершин v, w любого графа G , таких что w достижима из v в G , существует оптимальный путь из v в w , являющийся простым

Лемма (о подпутях). В основных допущениях для любого оптимального пути $v_1 - \dots - v_n$ любой его подпуть $v_i - \dots - v_j$, $1 \leq i \leq j \leq n$, также оптимален

Задача 1:

1. Докажите эти леммы
2. Верно ли, что в основных допущениях **каждый** оптимальный путь является простым?
3. Насколько существенно в этих леммах требование отсутствия циклов отрицательной стоимости?

Свойства графов

Дерево оптимальных путей из вершины v в графе G — это остовное дерево T графа G , в котором любой простой путь из v оптимален в G

Теорема (о дереве оптимальных путей). В основных допущениях для любой вершины v любого связного графа $G = (V, E)$ существует дерево оптимальных путей из v в G

Доказательство. Пронумеруем вершины графа: $V = \{v_1, \dots, v_n\}$

Построим последовательность деревьев

$T_0 = (V_0, E_0), T_1 = (V_1, E_1), \dots, T_n = (V_n, E_n)$ следующего вида:

1. Каждое дерево T_i является подграфом G и подграфом T_{i+1}
2. В каждом дереве T_i содержится соответствующая вершина v_i , если считать, что $v_0 = v$
3. Каждый простой путь из v в каждом дереве T_i является оптимальным в G

Очевидно, что если получится построить такую последовательность, то T_n — искомое дерево

Свойства графов

Доказательство.

$T_0 = (\{v\}, \emptyset)$ — очевидно, что это дерево подходит в качестве T_0

Для уже построенного дерева $T_{i-1} = (V_{i-1}, E_{i-1})$, $1 \leq i \leq n$, зададим $T_i = (V_i, E_i)$ так

Если $v_i \in V_{i-1}$, то $T_i = T_{i-1}$

Иначе $v_i \notin V_{i-1}$, и тогда устроим дерево T_i следующим образом

По лемме о простых путях, существует оптимальный простой путь из $v = w_1$ в $v_i = w_k$:

$$w_1 - w_2 - \dots - w_k$$

Пусть m — наибольший номер из $\{1, \dots, k\}$, такой что $w_m \in V_{i-1}$

Примем $V_i = V_{i-1} \cup \{w_{m+1}, \dots, w_k\}$ и

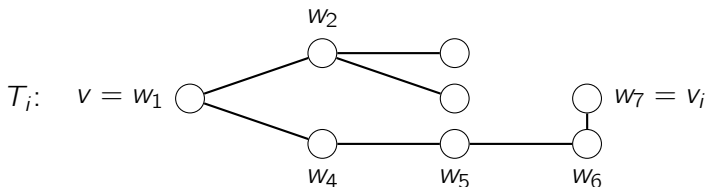
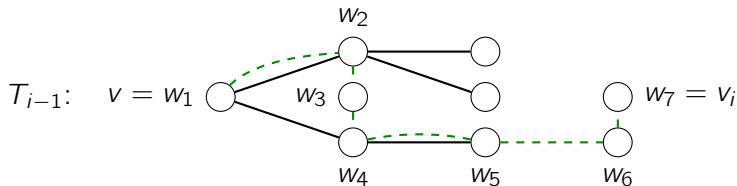
$$E_i = E_{i-1} \cup \{(w_m, w_{m+1}), (w_{m+1}, w_{m+2}), \dots, (w_{k-1}, w_k)\}$$

Свойства графов

Доказательство.

$v = w_1 - w_2 - \dots - w_k = v_i$ — оптимальный путь в G ; $m = \max(\ell \mid w_\ell \in V_{i-1})$;
 $V_i = V_{i-1} \cup \{w_{m+1}, \dots, w_k\}$; $E_i = E_{i-1} \cup \{(w_m, w_{m+1}), (w_{m+1}, w_{m+2}), \dots, (w_{k-1}, w_k)\}$

Иллюстрация:



Свойства графов

Доказательство.

$v = w_1 - w_2 - \dots - w_k = v_i$ — оптимальный путь в G ; $m = \max(\ell \mid w_\ell \in V_{i-1})$;
 $V_i = V_{i-1} \cup \{w_{m+1}, \dots, w_k\}$; $E_i = E_{i-1} \cup \{(w_m, w_{m+1}), (w_{m+1}, w_{m+2}), \dots, (w_{k-1}, w_k)\}$

По построению, T_{i-1} — подграф T_i и T_i — подграф G

T_i — дерево, так как этот граф связан и

$$|E_i| = |E_{i-1}| + k - m = |V_{i-1}| - 1 + k - m = |V_i| - 1$$

Осталось показать, что все простые пути из v в T_i оптимальны

Предположим от противного, что это не так: в T_i существует неоптимальный путь P из v

По построению (по предположению индукции), P не содержится в T_{i-1}

Значит, P оканчивается в вершине $w_\ell \in \{w_{m+1}, \dots, w_k\}$

Подпуть $w_1 - \dots - w_m$ пути P (содержится в T_i и) оптимален, а значит, по **лемме о подпутях** путь $w_m - \dots - w_\ell$ неоптимален

Но тогда по **лемме о подпутях** неоптимален и путь

$w_1 - \dots - w_m - \dots - w_\ell - \dots - w_k$ (*противоречие*) ▼

Маршрутизация и деревья оптимальных путей

Пусть в узле v вычислено какое-либо дерево T_u оптимальных путей из u в G

Тогда записью $table_v(u)$ обозначим родителя вершины v в T_u

Отображение $table_v$ можно считать таблицей маршрутизации узла v , и процедуру **продвижения пакета** с адресатом w в узле v можно устроить так:

- ▶ Если $v = w$, то продвижение завершено, доставить пакет
- ▶ Иначе отправить пакет узлу $table_v[w]$

Но в разных узлах могут быть сохранены разные деревья T_u : **не случится ли так, что из-за этого пакет будет бесконечно передаваться по циклу и не дойдёт до адресата?**

Маршрутизация и деревья оптимальных путей

Последовательность узлов сети v_1, \dots, v_k назовём

- ▶ **путём доставки** из узла v для адресата w , если верно следующее:
 - ▶ $v_1 = v, v_k = w$
 - ▶ $\forall i \in \{1, \dots, k-1\} : table_{v_i} = v_{i+1}$
- ▶ **циклом** для адресата w , если верно следующее:
 - ▶ все узлы v_i отличны от w
 - ▶ $\forall i \in \{1, \dots, k-1\} : table_{v_i}(w) = v_{i+1}$
 - ▶ $table_{v_n}(w) = v_1$

Будем говорить, что таблицы маршрутизации узлов сети

- ▶ **гарантируют доставку пакета** адресату w , если для любого узла v существуют путь доставки из v для w
- ▶ **содержат цикл**, если существует цикл для адресата w

Лемма (об ациклических таблицах). Таблицы маршрутизации узлов сети гарантируют доставку пакета адресату \Leftrightarrow в этих таблицах нет циклов для w

Маршрутизация и деревья оптимальных путей

Доказательство.

(\Rightarrow) Пусть в таблицах маршрутизации содержится цикл v_1, \dots, v_k для адресата w

Тогда $\forall i \in \{1, \dots, k\} : table_{v_i}(w) \in \{v_1, \dots, v_k\}$

При этом $w \notin \{v_1, \dots, v_k\}$

Значит, пакет, поступивший в узел v_1 , не будет доставлен адресату w

(\Leftarrow) Пусть таблицы маршрутизации не гарантируют доставку пакета адресату w

Тогда существует узел v , из которого не существует пути доставки для адресата w

Следовательно, среди значений

$v_1 = v, v_2 = table_{v_1}(w), v_3 = table_{v_2}(w), \dots$ не встречается w

Так как в сети содержится конечное число узлов, то среди значений v_1, v_2, v_3, \dots есть по крайней мере два одинаковых: $v_i = v_j$, где $i < j$

Тогда v_1, v_2, \dots, v_j — цикл для w ▼

Маршрутизация и деревья оптимальных путей

Задача 2.

Допустим, что

- ▶ топология сети может произвольно изменяться во время выполнения сети,
- ▶ в начале выполнения и после каждого изменения сохраняются связность графа топологии и основные допущения и
- ▶ между изменением топологии и следующим за ним продвижением пакета узлом обязательно выполняется вычисление таблиц в этом узле

Докажите, что в таких условиях никакой распределённый алгоритм (с асинхронным обменом сообщениями) не способен гарантировать доставку пакетов всем адресатам