

Распределённые алгоритмы

mk.cs.msu.ru → Лекционные курсы → Распределённые алгоритмы

Блок 19

Алгоритм Туэга

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2025, февраль–май

Задача и допущения

Вернёмся к задаче вычисления таблиц маршрутизации: требуется в каждом узле s р.с. для каждого узла-адресата d вычислить значение $table_s[d]$ следующего узла оптимального пути из s в d

Для определённости положим, что если $s = d$ или d недостижимо из s , то $table_s[d] = \perp$

Будем использовать такие **основные допущения**:

1. Граф топологии, для которого решается задача, — это взвешенный граф $\Gamma = (G, \omega)$, не содержащий циклов **неположительного** веса
2. $G = (V, E)$
3. Все описания и утверждения формулируются для произвольного Γ указанного выше вида
4. Каждый узел v знает
 - ▶ имена всех узлов и их порядок (v_1, \dots, v_n), одинаковый для всех узлов
 - ▶ имена своих соседей: множество $Neigh_v$
 - ▶ веса каналов, соединяющих его с соседями

Обсуждение алгоритма Туэга принято начинать с его упрощённой версии (для лучшего восприятия)

Упрощённый алгоритм Туэга

Основные отличия упрощённого алгоритма Туэга от алгоритма Флойда-Уоршелла:

- ▶ Переменные и операции над ними «разнесены» по узлам сети: значение $\rho[s, d] = \rho_s[d]$ вычисляется в ячейке $r_s[d]$ массива r_s узла s
- ▶ Перед обновлением значения переменной r_s узел s получает всю необходимую информацию из сообщений
- ▶ Информация об опорном узле v отправляется всем узлам при помощи команды «`broadcast`»:
 - ▶ при выполнении `broadcast(x)` отправляются сообщения x со всеми узлами, кроме узла-отправителя, в качестве адресатов
- ▶ В узле s вычисляются не только требуемые веса (ρ_s в массиве r_s), но и значения таблицы маршрутизации ($table_s$ в массиве τ_s)

Упрощённый алгоритм Туэга

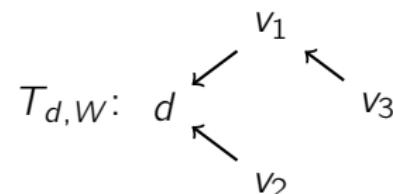
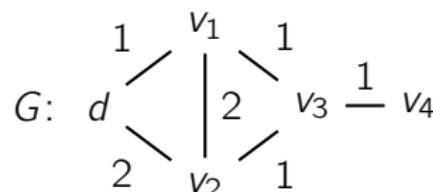
Оптимальным W -путём из s в d назовём W -путь из s в d наименьшего веса среди всех W -путей из s в d

Лемма (об отсутствии циклов). Пусть:

- ▶ Справедливы основные допущения
- ▶ Заданы множество узлов W и узел d
- ▶ Если $\rho^W[s, d] < \infty$ и $s \neq d$, то $\tau_s^W[d]$ — имя узла, следующего за s в каком-либо оптимальном W -пути из s в d
- ▶ $T_{d, W} = (V_{d, W}, E_{d, W})$ — орграф, в котором:
 - ▶ $V_{d, W} = \{s \mid s \in V, \rho^W[s, d] < \infty\}$
 - ▶ $E_{d, W} = \{(s, v) \mid (s, v) \in V_{d, W}^2, s \neq d, \tau_s^W[d] = v\}$

Тогда $T_{d, W}$ — дерево с корнем (стоком) d

Иллюстрация (для $W = \{v_1, v_2\}$):



Упрощённый алгоритм Туэга

Доказательство.

По **лемме о подпутях**, если для узла s верно $\rho^W[s, d] < \infty$ и $s \neq d$, то $\tau_s^W[d] \neq \perp$

Следовательно, для каждого узла $s \in V_{d,W} \setminus \{d\}$ существует (единственный) узел $v \in V_{d,W}$, такой что $\tau_s^W[d] = v$

То есть для каждого $s \in V_{d,W} \setminus \{d\}$ в $E_{d,W}$ существует единственная дуга вида $s \rightarrow \underline{}$

При этом по построению из d в $T_{d,W}$ не исходит ни одной дуги

Следовательно, $|E_{d,W}| = |V_{d,W}| - 1$

Упрощённый алгоритм Туэга

Доказательство.

Предположим от противного, что в $T_{d,W}$ не дерево

Из этого, соотношения $|E_{d,W}| = |V_{d,W}| - 1$

и того, что в $T_{d,W}$ из каждой вершины исходит не более одной дуги,
следует, что в $T_{d,W}$ существует цикл
(для ясности — $v_1 \rightarrow \dots \rightarrow v_k \rightarrow v_1$)

Для любого канала $(x, y) \in E_{d,W}$ верно $\rho^W(x, d) = \omega(x, y) + \rho^W(y, d)$

Значит, верно равенство

$$\rho^W(v_1, d) = \omega(v_1, v_2) + \omega(v_2, v_3) + \dots + \omega(v_{k-1}, v_k) + \omega(v_k, v_1) + \rho^W(v_1, d)$$

Но тогда $\omega(v_1, v_2) + \omega(v_2, v_3) + \dots + \omega(v_{k-1}, v_k) + \omega(v_k, v_1) = 0$ — в $T_{d,W}$
содержится цикл веса 0, что *противоречит* основному допущению 1

Следовательно, $T_{d,W}$ — дерево ▼

Упрощённый алгоритм Туэга

Упрощённый алгоритм Туэга: каждому узлу s отвечает следующий код

1. $W_s := \emptyset;$
2. Для каждого $v \in V$:
 - 2.1 Если $v = s$: $r_s[v] := 0$; $\tau_s[v] := \perp$;
 - 2.2 Если $v \in \text{Neigh}_s$: $r_s[v] := \omega(s, v)$; $\tau_s[v] := v$;
 - 2.3 Иначе: $r_s[v] := \infty$; $\tau_s[v] := \perp$;
3. Пока $W_s \neq V$:
 - 3.1 Произвольно выбрать $v \in V \setminus W_s$ одинаково для всех узлов (опорный узел v)
 - 3.2 Если $v = s$, то $\text{broadcast}(r_s)$, иначе $\text{receive}_v(r_v)$
 - 3.3 Для всех $d \in V$:
 - 3.3.1 Если $r_s[v] + r_v[d] < r_s[d]$: $r_s[d] := r_s[v] + r_v[d]$; $\tau_s[d] := \tau_s[v]$;
 - 3.4 $W_s := W_s \cup \{v\}$;
4. Выдать ответ: $\rho_s = r_s$, $\text{table}_s = \tau_s$

Упрощённый алгоритм Туэга

Теорема (корректность). Любое вычисление упрощённого алгоритма Туэга в основных допущениях конечно, в последней его конфигурации для любых узлов s, d верно

- ▶ $r_s[d] = \rho[s, d]$ и
- ▶ если $\rho[s, d] < \infty$ и $s \neq d$, то $\tau_s[d] = \text{table}_s[d]$,

и если граф G связен, то таблицы τ_s в последней конфигурации гарантируют доставку пакетов каждому адресату

Доказательство.

Завершаемость и корректность значений $r_s[d]$ следуют из **корректности алгоритма Флойда-Уоршелла** и одинаковости устройства вычислений этих двух алгоритмов

Корректность значений $\tau_s[d]$ следует из того, что они изменяются непосредственно после изменения соответствующих $r_s[d]$, и при изменении в $\tau_s[d]$ присваивается следующий узел в пути веса $r_s[d]$. Гарантия доставки пакетов обеспечивается **леммой об отсутствии циклов** и **леммой об ациклических таблицах** ▼

Алгоритм Туэга

К сожалению, алгоритм Туэга, согласно рассказанному сейчас в лекциях, **нереалистичен** из-за команды `broadcast`

Осмыслить эту команду как **реалистичную** помогут обсуждающиеся позже **волновые алгоритмы**, но пока не будем использовать их в аргументации

Тогда, чтобы избежать **нереалистичности**, можно использовать следующее наблюдение

Если при выборе опорного узла v верно $r_s[v] = \infty$, то для каждого узла d верно $r_s[v] + r_v[d] = \infty \geq r_s[d]$, и в (3) при выборе такого v значения r_s и τ_s не изменяются

Поэтому выполнение $\text{broadcast}(r_s)$ в (3.2) можно заменить на распространение r_s по каналам текущего дерева $\mathcal{T}_v = T_{v, W_v}$ от корня:

- ▶ Узел v отправляет r_v всем своим детям в \mathcal{T}_v
- ▶ Каждый другой узел \mathcal{T}_v , получив r_v , пересыпает это значение всем своим детям в \mathcal{T}_v

Алгоритм Туэга

В начале (3) с опорной вершиной v каждый узел s , для которого верно $d_s[v] < \infty$, знает своего родителя в \mathcal{T}_v , но не знает детей для отправки r_v

Чтобы каждый узел узнал о своих детях в \mathcal{T}_v , достаточно «заставить» каждый узел w послать всем своим соседям сообщение о том, является ли этот сосед родителем узла w в \mathcal{T}_v

Получив такие сообщения от всех соседей, каждый узел s узнает о всех своих детях в \mathcal{T}_v

Алгоритм Туэга

Инициализация в узле s ($Init_s$):

1. $W_s := \emptyset;$
2. Для всех $d \in V$:
 - 2.1 Если $d = s$:
 - 2.1.1 $r_s[d] := 0;$
 - 2.1.2 $\tau_s[d] := \perp;$
 - 2.2 Если $d \in Neigh_s$:
 - 2.2.1 $r_s[d] := \omega(s, d);$
 - 2.2.2 $\tau_s[d] := d;$
 - 2.3 Иначе:
 - 2.3.1 $r_s[d] := \infty;$
 - 2.3.2 $\tau_s[d] := \perp;$

Алгоритм Туэга

Выявление детей в узле s для опорного узла v ($Children_s(v)$):

1. Для всех $w \in Neigh_s$:
 - 1.1 $send_w(\mathbf{par}, v, (\tau_s[v] = w))$
2. $N_v := 0$; $Ch_s := \emptyset$;
3. Пока $N_v < |Neigh_v|$:
 - 3.1 $receive_w(\mathbf{par}, \underline{v}, b)$ от любого $w \in Neigh_v$
 - 3.2 Если b : $Ch_s := Ch_s \cup \{w\}$;
 - 3.3 $N_v := N_v + 1$;

Пояснение

Сообщение $(\mathbf{par}, v, \mathbf{t})$ от s к w означает, что w является родителем s в \mathcal{T}_v , а $(\mathbf{par}, v, \mathbf{f})$ — что не является

Дети узла s в \mathcal{T}_v — это все соседи s , которые отправили ему сообщение $(\mathbf{par}, v, \mathbf{t})$

Алгоритм Туэга

Уточнение таблиц маршрутизации в узле s для опорного узла v ($Recompute_s(v)$):

1. Если $r_s[v] < \infty$:

1.1 Если $s \neq v$:

1.1.1 $receive_{\tau_s[v]}(\mathbf{tab}, v, r_v)$

1.2 Для всех $w \in Ch_s$:

1.2.1 $send_w(\mathbf{tab}, v, r_v)$

1.3 Для всех $d \in V$, таких что $r_s[v] + r_v[d] < r_s[d]$:

1.3.1 $r_s[d] := r_s[v] + r_v[d]$;

1.3.2 $\tau_s[d] := \tau_s[v]$;

Пояснение

Узел v располагает таблицей r_v и отсылает её всем детям в T_v

Каждый другой узел получает r_v от родителя (и теперь может её использовать) и пересыпает детям

Следовательно, каждый узел из T_v рано или поздно выполнит (1.3), располагая таблицей r_v

Алгоритм Туэга

Алгоритм Туэга (полный): каждому узлу s отвечает следующий код

1. $Init_s$
2. Пока $W_s \neq V$:
 - 2.1 Произвольно выбрать $v \in V \setminus W_s$ одинаково для всех узлов (опорный узел)
 - 2.2 $Children_s(v)$
 - 2.3 $Recompute_s(v)$
 - 2.4 $W_s := W_s \cup \{v\}$;

Алгоритм Туэга

Теорема (корректность). Любое вычисление алгоритма Туэга в основных допущениях конечно, в последней его конфигурации для любых узлов s, d верно

- ▶ $r_s[d] = \rho[s, d]$ и
- ▶ если $\rho[s, d] < \infty$ и $s \neq d$, то $\tau_s[d] = \text{table}_s[d]$,

и если граф G связан, то таблицы τ_s в последней конфигурации гарантируют доставку пакетов каждому адресату

Доказательство. Следует из

- ▶ корректности упрощённого алгоритма Туэга,
- ▶ корректности процедуры $\text{Children}_s(v)$ (см. ниже): после её выполнения Ch_s — это множество всех детей s в T_v — и
- ▶ пояснений об особенностях выполнения кода, сделанных непосредственно после кода

Корректность процедуры $\text{Children}_s(v)$ следует из определения дерева T_v и того, что детьми узла s в T_v являются те и только те узлы w , для которых w — сосед s и s — родитель w ▼

Алгоритм Туэга

Теорема (сложность). Алгоритм Туэга имеет коммуникационную сложность $O(nm)$ и битовую сложность $O(n^3k)$, где $n = |V|$, $m = |E|$ и k — число битов, использующееся для записи одного имени вершины и одного веса пути. Кроме того, при выполнении алгоритма

- ▶ в каждый канал отправляется $O(n)$ сообщений и $O(n^2k)$ битов и
- ▶ для хранения всех текущих значений в узле используется $O(nk)$ битов памяти

Доказательство.

Для каждой опорной вершины v в каждый канал отправляются

- ▶ 2 сообщения типа **par** (по одному в каждом направлении) и
- ▶ не более 1 сообщения типа **tab**

Сообщение типа **par** содержит $O(k)$ битов, типа **tab** — $O(nk)$ битов

Всего перебирается n опорных вершин

Из этого следуют оценки сложности для канала

Алгоритм Туэга

Теорема (сложность). Алгоритм Туэга имеет коммуникационную сложность $O(pt)$ и битовую сложность $O(n^3k)$, где $n = |V|$, $m = |E|$ и k — число битов, использующееся для записи одного имени вершины и одного веса пути. Кроме того, при выполнении алгоритма

- ▶ в каждый канал отправляется $O(n)$ сообщений и $O(n^2k)$ битов и
- ▶ для хранения всех текущих значений в узле используется $O(nk)$ битов памяти

Доказательство.

Самые объёмные единовременно хранящиеся данные: r_s , r_v , τ_s — содержат $O(n)$ значений размера k , откуда следует оценка использующихся битов памяти $O(nk)$

Коммуникационная сложность получается из сложности $O(n)$ для одного канала домножением на число каналов m

Алгоритм Туэга

Теорема (сложность). Алгоритм Туэга имеет коммуникационную сложность $O(nm)$ и битовую сложность $O(n^3k)$, где $n = |V|$, $m = |E|$ и k — число битов, использующееся для записи одного имени вершины и одного веса пути. Кроме того, при выполнении алгоритма

- ▶ в каждый канал отправляется $O(n)$ сообщений и $O(n^2k)$ битов и
- ▶ для хранения всех текущих значений в узле используется $O(nk)$ битов памяти

Доказательство.

Битовая сложность

При выполнении алгоритма суммарно в каналы отправляется $2nm$ сообщений типа **par**: оценка для одного канала, помноженная на число каналов

Размер каждого такого сообщения — $O(k)$

Суммарное число битов в таких сообщениях: $O(nmk)$

С учётом известного неравенства $m \leq n^2$, $O(nmk)$ означает и $O(n^3k)$

Алгоритм Туэга

Теорема (сложность). Алгоритм Туэга имеет коммуникационную сложность $O(nm)$ и битовую сложность $O(n^3k)$, где $n = |V|$, $m = |E|$ и k — число битов, использующееся для записи одного имени вершины и одного веса пути. Кроме того, при выполнении алгоритма

- ▶ в каждый канал отправляется $O(n)$ сообщений и $O(n^2k)$ битов и
- ▶ для хранения всех текущих значений в узле используется $O(nk)$ битов памяти

Доказательство.

Битовая сложность

При выполнении алгоритма суммарно в каналы отправляется не более n^2 сообщений типа **tab**: n опорных вершин, и для каждой — не более чем $(n - 1)$ (количество рёбер в дереве T_v)

Размер каждого такого сообщения — $O(nk)$

Суммарное число битов: $O(n^2nk) = O(n^3k)$

Сложив оценки для **par** и типа **tab**, получим оценку $O(n^3k)$ ▼

Алгоритм Туэга

Д.з. 1. Зачем в алгоритме Туэга во всех сообщениях передаётся имя v опорной вершины? Можно ли не передавать v хотя бы в каких-нибудь из сообщений? Если нет, то почему алгоритм перестанет быть корректным?

Д.з. 2. Можно ли модифицировать алгоритм так, чтобы не передавать сообщения (par, v, f) , мотивируя это тем, что узел может по умолчанию полагать, что у него нет детей? Если нет, то почему алгоритм перестанет быть корректным?

Алгоритм Туэга

Алгоритм Туэга

- ▶ относительно прост,
- ▶ имеет невысокую сложность и
- ▶ вычисляет веса всех оптимальных путей

Но он имеет и **недостатки**:

- ▶ При изменении топологии сети требуется пересчитывать всё «с ноля»
- ▶ Должны быть заранее согласованы множество V и порядок выбора опорных вершин
- ▶ При пересчете значений r_s используется таблица r_v , которую могут не знать ни узел s , ни его соседи, и в худшем случае следует распространить эту таблицу по всей сети