

# ЗАДАЧА О БЫСТРЕЙШЕМ ПЕРЕМЕЩЕНИИ НА ГРАФЕ С ПЕРЕМЕННЫМИ ЗАДЕРЖКАМИ

И.В.Коннов

Казань, эл.почта: *konn-igor@ya.ru*

Для динамической задачи о быстрейшем перемещении между заданными вершинами графа с произвольными функциями задержек предложен и обоснован модифицированный алгоритм Дейкстры.

*Ключевые слова:* Экстремальные задачи на графах, задача о быстрейшем перемещении, переменные задержки дуг, алгоритм типа Дейкстры.

## 1 Постановка задачи

Напомним, что граф задается множеством вершин  $V$ , которое будет считаться конечным, множеством дуг  $\mathcal{A}$  и отображением, ставящим в соответствие паре вершин  $i$  и  $j$  элемент  $a \in \mathcal{A}$ . Если значение отображения для  $i$  и  $j$  непусто, то определена дуга  $a = (i, j)$ , для которой вершина  $i$  является исходящей (начальной), а  $j$  – входящей (конечной). Если значение отображения для  $i$  и  $j$  пусто, то вершины  $i$  и  $j$  не связаны дугой. Последовательность дуг

$$a_1, a_2, a_3, \dots, a_k,$$

где  $a_1 = (i_1, i_2)$ ,  $a_k = (i_k, i_{k+1})$ , и любые две соседние дуги имеют смежную вершину, называется цепью, соединяющей вершины  $i_1$  и  $i_{k+1}$ . Граф называется связным, если любые две его вершины можно соединить цепью. Путем называется цепь, которую можно пройти от начальной до конечной вершины в направлении дуг. Граф называется связным по путям, если любые две его вершины можно соединить путем. Циклом называется замкнутая цепь, у которой начальная и конечная вершины совпадают. Замкнутый путь, т.е. цикл, который можно пройти в направлении дуг, называется контуром. Деревом называется связный граф без циклов.

С графами связано множество прикладных экстремальных задач. Одной из наиболее известных и часто используемых является задача о кратчайшем связывающем пути в графе. Всем дугам графа приписывают значения, называемые весами. Задача будет состоять в том, чтобы для заданного графа и пары его вершин  $v$  и  $w$  найти путь с минимальным суммарным весом дуг из  $v$  в  $w$ . Веса дуг могут иметь различный содержательный смысл. Если вес дуги  $a \in \mathcal{A}$  – это время прохождения дуги (задержка), то получаем задачу о быстрейшем перемещении между заданными вершинами графа. Наиболее изученной является задача о кратчайшем пути с постоянными весами.

В настоящей работе изучается задача о быстрейшем перемещении между заданными вершинами графа с переменными задержками дуг. Каждой дуге графа  $a = (i, j) \in \mathcal{A}$  приписывается неотрицательная функция  $t_{ij}(d_i)$  – время прохождения дуги  $a$ , если движение из вершины  $i$  начато в момент времени  $d_i$ . Граф предполагается связным по путям. Требуется для заданной пары вершин  $v$  и  $w$  и начального момента времени  $d'_0$  найти путь с минимальным временем перемещением по дугам из  $v$  в  $w$ . Для упрощения далее считаем, что  $d'_0 = 0$ .

Сложность решения этой задачи зависит от применяемых дополнительных условий. В частности, можно допустить или разрешить простои в движении в вершинах, причем задача без простоев гораздо сложнее. Можно ввести дополнительные условия на функцию задержки. В частности, в [1] эта задача изучалась при условии монотонности прибытия, что автоматически влечет невыгодность простоев в вершинах. Там же в этом случае предложен и обоснован простой алгоритм Дейкстры. Однако во многих приложениях функции задержки могут быть достаточно произвольными, не обладать свойствами монотонности. Тогда алгоритм Дейкстры неприменим при требовании отсутствия простоев, что иллюстрирует следующий пример.

**Пример 1** Граф содержит четыре вершины, которые связаны пятью дугами, как показано на рис. 1. Пусть  $v = v_1$  и  $w = v_4$ ,  $t_{v_1, v_2} \equiv 2$ ,  $t_{v_1, v_3} \equiv 5$ ,  $t_{v_2, v_3} \equiv 2$ ,  $t_{v_2, v_4} \equiv 7$ ,

$$t_{v_3, v_4}(d_{v_3}) = \begin{cases} 4, & \text{если } 4 \leq d_{v_3} < 5, \\ 2, & \text{если } 5 \leq d_{v_3} \leq 6. \end{cases}$$

Тогда путь  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$  дает 8, причем подпуть  $v_1 \rightarrow v_2 \rightarrow v_3$  – быстрейший, путь  $v_1 \rightarrow v_2 \rightarrow v_4$  дает 9, но путь  $v_1 \rightarrow v_3 \rightarrow v_4$  дает 7. Это быстрейший путь, хотя его часть  $v_1 \rightarrow v_3$  не наилучшая.

Отметим, что произвольные функции задержки могут быть связаны не только с различными режимами работы дуг по времени, но и с использованием различных средств перемещения по дугам.

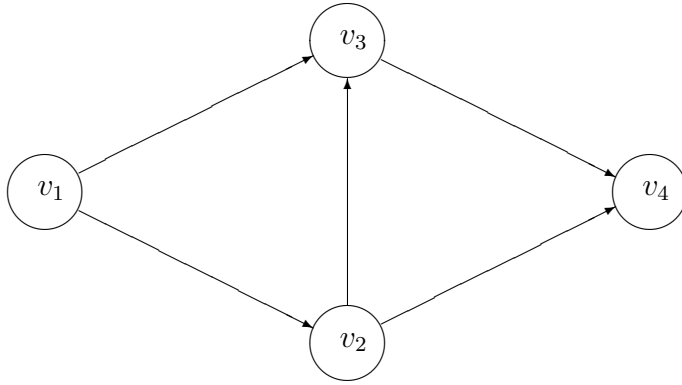


Рис. 1: Наилучший путь содержит невыгодный подпуть

## 2 Алгоритм и его обоснование

Для задачи о быстрейшем перемещении между заданными вершинами  $v$  и  $w$  графа с возможностью простоев в движении в вершинах опишем модифицированный алгоритм Дейкстры. Функции задержек предполагаются произвольными неотрицательными, полунепрерывными и ограниченными снизу функциями, тогда  $d_i + t_{ij}(d_i) \rightarrow +\infty$ , если  $d_i \rightarrow +\infty$ .

**Алгоритм.** Вначале полагаем  $V' = \emptyset$ ,  $V'' = V$ ,  $H' = \emptyset$ ,  $t'_v = 0$ ,  $t'_j = +\infty$  для  $j \in V'' \setminus \{v\}$ .

На очередной итерации выбираем элемент  $k$  из  $V''$ , такой что

$$t'_k = \min_{j \in V''} t'_j,$$

полагаем  $V' = V' \cup \{k\}$ ,  $V'' = V'' \setminus \{k\}$ . Если  $V' \neq \{k\}$ , то также полагаем  $H' = H' \cup \{(i(k), k)\}$ . Если  $k = w$ , то останов. Далее для каждой вершины  $s \in V''$ ,  $(k, s) \in \mathcal{A}$  вычисляем

$$d'_{ks} = \arg \min \{d_k + t_{ks}(d_k) \mid d_k \geq t'_k\},$$

полагаем  $t''_s = d'_{ks} + t_{ks}(d'_{ks})$ . Если  $t''_s < t'_s$ , то полагаем  $t'_s = t''_s$ ,  $i(s) = k$ .

В этом алгоритме величина  $t'_i$  определяет текущее наименьшее время перемещения из вершины  $v$  в  $i$ ,  $V'$  определяет текущее множество вершин, для которых точное наименьшее время перемещения вычислено,  $V''$  определяет текущее оставшееся множество вершин,  $H'$  определяет дерево дуг с наименьшим временем

перемещения. Если выполнять останов по условию  $V'' = \emptyset$ , то алгоритм найдет быстрые перемещения между вершиной  $v$  и всеми остальными вершинами графа.

**Теорема.** Алгоритм для любого  $i$  из  $V'$  определяет точное наименьшее время перемещения  $t'_i$  между вершиной  $v$  и вершиной  $i$ , в то же время  $H'$  определяет для них дуги с наименьшим временем перемещения.

Для обоснования будет достаточно показать, что при определении индекса  $k$  на любой итерации величина  $t'_k$  определяет точное наименьшее время перемещения между вершиной  $v$  и вершиной  $k$ .

Вначале при  $k = v$  это очевидно, поскольку  $t'_v = 0$ ,  $t'_j = +\infty$  для  $j \in V'' \setminus \{v\}$ . Следующим будет выбран индекс  $k$ , такой что  $(v, k) \in \mathcal{A}$  и

$$t'_k = t_{vk}(d'_{vk}) \leq t'_j = t_{vj}(d'_{vj}), \quad \forall (v, j) \in \mathcal{A}.$$

Это значит, что индекс  $k$  на этой итерации определяет вершину и точное наименьшее время перемещения  $t'_k$  между вершиной  $v$  и вершиной  $k$ , поскольку остальные величины не могут быть меньше.

Предположим, что после конечного числа итераций и полученных текущих множеств  $V'$  и  $V''$  для любого  $i$  из  $V'$  величина  $t'_i$  определяет точное наименьшее время перемещения между вершиной  $v$  и вершиной  $i$ . Пусть выбран индекс  $k \in V''$  на итерации, но величина  $t'_k$  не является наименьшим временем перемещения между вершиной  $v$  и вершиной  $k$ . Это значит, что существует последовательность вершин

$$i_1, i_2, \dots, i_{l-1}, i_l, \dots, i_t,$$

такая что  $i_1 = v$ ,  $i_t = k$  и соответствующая последовательность дуг

$$(i_{s-1}, i_s) \in \mathcal{A}, \quad s = \overline{2, t}.$$

Теперь если определить  $\tilde{t}_v = t'_v = 0$ ,

$$\tilde{t}_{i_s} = \min\{d_{i_{s-1}, i_s} + t_{i_{s-1}, i_s}(d_{i_{s-1}}) \mid d_{i_{s-1}} \geq \tilde{t}_{i_{s-1}}\}, \quad s = \overline{2, t},$$

то  $\tilde{t}_k < t'_k$ . Таким образом, величина  $\tilde{t}_{i_s}$  определяет наименьшее время перемещения между вершиной  $v$  и вершиной  $i_s$  для выбранного пути, который лучше полученного в алгоритме.

Отметим, что при этом найдется пара вершин  $i_{l-1} \in V'$ ,  $i_l \notin V'$ , поскольку  $v = i_1 \in V'$ ,  $k = i_t \notin V'$ , возможно, что  $l = t$ . Пусть  $l$  – наименьший номер вершины в пути с таким свойством. По построению пути имеем  $\tilde{t}_{i_l} \leq \tilde{t}_{i_l} < t'_k$ , но  $i_l \notin V'$ . С другой стороны,  $t'_{i_{l-1}} \leq \tilde{t}_{i_{l-1}}$ , тогда  $t'_{i_l} \leq \tilde{t}_{i_l}$  для текущей итерации из-за совпадения правила выбора момента движения. Действительно, индекс  $i_{l-1}$

по алгоритму должен был попасть в  $V'$  раньше текущей итерации, тогда же было вычислено значение  $t'_{i_l}$  – текущее наименьшее время перемещения из вершины  $v$  в  $i_l$ , поскольку  $(i_{l-1}, i_l) \in \mathcal{A}$ , а оно не могло уменьшиться. Отсюда следует  $t'_{i_l} < t'_k$  для текущей итерации, при этом  $i_l \in V''$ . Однако это противоречит правилу выбора индекса  $k$  на итерации.

### 3 Обсуждение

Хорошо известно, что существенные колебания параметров потока достаточно типичны для различных транспортных сетей и сетей связи. Поэтому постановка задачи о быстрейшем перемещении с произвольными функциями задержки гораздо лучше будет подходить в качестве математической модели реальной сети. Более того, такая постановка удобна даже при стационарном поведении транспортного потока, когда перемещение по дуге может происходить в несколько выбранных дискретных моментов времени, по расписанию. Если все транспортные средства на дуге одностипны, то в функцию задержки можно включить и время ожидания рейса.

**Пример 2** Пусть в промежутке времени  $[0, 4]$  по дуге  $(i, j)$  выполняются два рейса по 1 часу: в 1 час и в 4 часа. Тогда можно определить функцию задержки

$$t_{ij}(d_i) = \begin{cases} (1 - d_i) + 1, & \text{если } 0 \leq d_i \leq 1, \\ (4 - d_i) + 1, & \text{если } 1 < d_i \leq 4. \end{cases}$$

Здесь движение отождествляется с выбором рейса.

Кроме того, по одной дуге в разное время могут выполняться рейсы разными транспортными средствами, с разным временем рейса. Тогда удобнее отметить все дискретные моменты времени начала рейсов с указанием реального времени задержки, а затем построить интерполяционную кусочно-линейную функцию задержки для этой дуги.

Такой подход позволяет учитывать также другие факторы при выборе маршрута перемещения, например, стоимость и надежность. В частности, при установке ограничений сверху на стоимость и снизу на надежность при перемещении по дуге следует исключить не удовлетворяющие им транспортные средства и скорректировать функцию задержки.

Требование полунепрерывности снизу функций задержек можно снять, достаточно, чтобы задача выбора числа  $d'_{ks}$  на каждой итерации имела решение.

### Список литературы

- [1] Гимади Э.Х., Глебов Н.И. Дискретные экстремальные задачи принятия решений. – Новосибирск: НГУ, 1991.