О задаче поиска оптимальных маршрутов

А. Н. Баушев
Петербургский государственный университет путей сообщения Александра I (ПГУПС)
banban2008@yandex.ru

О. Л. Семенова
Санкт-Петербургский государственный университет (СПбГУ)
o_semenova@mail.ru

И. Т. Утепбергенов[1], А. Т. Ахмедиярова[2]
Институт информационных и вычислительных технологий КН МОН РК (ИИВТ)
[1] i.utepbergenov@gmail.com, [2]Akhmediyarova@turan-edu.kz

***Аннотация*. В докладе описывается BFS-алгоритм поиска оптимального пути во взвешенном графе при ограничениях, задаваемых конечным числом неравенств, накладываемых на дополнительные веса искомого пути. Оптимизационная задача при таких ограничениях становится NP-сложной, а предложенный нами алгоритм её решения – псевдо-полиномиальным. По существу, наш алгоритм представляет собой модификацию хорошо известного алгоритма Беллмана–Мура [1], в котором операция релаксации заменяется операцией фильтрации совокупности допустимых путей. Благодаря тому, что в типичных ситуациях размер множества Парето конечного множества точек существенно меньше, чем размер самого множества, предложенный алгоритм способен эффективно решать оптимизационную задачу на довольно больших сетях.**

Ключевые слова: алгоритм Беллмана–Мура; множество Парето; операция релаксации; составление расписаний

I. Введение

Предположим, что транспортная компания планирует работу транспортного средства на время T для перевозки грузов (или пассажиров) по некоторой сети дорог. В начальный момент рабочего периода средство находится в некотором узле s дорожной сети, а по истечении времени T должно быть возвращено в узел t (возможно, совпадающий с узлом s). При перевозке груза (пассажиров) между смежными узлами сети компания получает прибыль, которая является случайной величиной, имею щей распределение, определяемое данной парой узлов. Время, затрачиваемое на перемещение между смежными узлами, является неслучайной функцией этой (упорядоченной) пары узлов. Спрашивается, как спланировать работу транспортного средства на период T, чтобы математическое ожидание суммарной прибыли было максимальным?

Для построения математической модели задачи рассмотрим ориентированный граф $G = \langle V, A\rangle$ на множестве вершин V, представляющем множество узлов дорожной сети, и с множеством дуг A, представляющем дороги между смежными узлами. Для дуги $a \in A$ обозначим через $\tau(a)$ время, затрачиваемое на перемещение по этой дуге, а через $w(a)$ обозначим величину $-\mathbb{E}X(a)$, где $X(a)$ – прибыль, получаемая при перемещении транспортного средства по этой дуге. Пусть $\mathcal{M}(s,t)$ обозначает совокупность всех путей в графе G с начальной вершиной s и конечной вершиной t. Для пути $\mu \in \mathcal{M}(s,t)$ его веса $\tau(\mu)$ и $w(\mu)$ определяются как суммы соответствующих весов, входящих в него дуг.

В этих обозначениях математическую модель задачи можно записать в виде

$$w(\mu) \to min,\ \mu \in \mathcal{M}(s,t), \tau(\mu) \le T. \tag{1}$$

Мы рассмотрим обобщение задачи (1) на случай нескольких τ-весов:

$$w(\mu) \to min,\ \mu \in \mathcal{M}(s,t),$$
$$\tau_i(\mu) \le T_i,\ i = 1, \dots, N, \tag{2}$$

где $T_1, \dots T_N$ – заданные числа.

Задачи(1) и (2) изучались в [2] в связи с задачей составления расписаний, где для частного случая задачи (1), в котором $w \ge 0,\ \tau > 0,\ T > 0$, был предложен алгоритм построения её точного решения.

Мы рассматриваем общий случай, в котором N может быть произвольным, а веса и параметры $T_1, \dots, T_N$ могут иметь произвольные знаки. Достаточным условием корректной работы предлагаемого алгоритма являются условие отсутствия в графе G циклов неположительного веса хотя бы для одного из весов $\tau_1, \dots \tau_N$.

Отметим, что использование весов произвольного знака позволяет свести задачу

$$w(\mu) \to min,\ \mu \in \mathcal{M}(s,t),$$
$$T_i' \le \tau_i(\mu) \le T_i'',\ i = 1, \dots, N,$$

к задаче вида (2) путём введения дополнительных τ-весов.

С точки зрения теории сложности вычислений, задачи (1) и (2) принадлежат классу NP-трудных задач: нетрудно показать, что классическая задача о рюкзаке [3] сводима по Карпу к задаче (1). Однако для задач (1) и (2), так же как и для задачи о рюкзаке, существует простой, псевдо-

Работа поддержана грантом МОН КР (проект № AP05133699)

полиномиальный по сложности, алгоритм решения, который оказывается практически эффективным на больших сетях, если τ-веса в этих задачах целочисленные и параметры $T, T_1, \ldots, T_N$ не слишком велики.

Предлагаемый нами алгоритм решения задачи (2), по существу, представляет собой модификацию хорошо известного алгоритма Беллмана–Мура для нахождения путей минимального веса во взвешенном графе [1]. Модификация заключается в замене операции *релаксации* классического алгоритма операцией *фильтрации*, позволяющей значительно редуцировать совокупность путей, среди которых ищется решение задачи (2).

Пусть $(P, \preccurlyeq)$ – произвольное частично упорядоченное множество, $X \subset P$. Элемент $y \in X$ называется *минимальным элементом множества* X, если $\{x \in X : x \prec y\} = \emptyset$. В экономической литературе множество $Min(X)$ всех минимальных элементов множества X называют так же *множеством Парето* для множества X.

Отображение Φ, которое каждому подмножеству множества P ставит в соответствие множество его минимальных элементов, называется *фильтрацией* совокупности подмножеств *отношением частичного порядка* $\preccurlyeq$, а неподвижные точки этого отображения – *отфильтрованными* множествами.

Отметим, что если множество $X \subset \mathbb{R}^d$ получено процедурой случайного выбора из генеральной совокупности с непрерывным распределением и независимыми координатами, то математическое ожидание числа точек множества $Min(X)$ оценивается сверху величиной $O\left(log^{d-1}(\#X)\right)$ [4]. Поэтому в типичных ситуациях $\#Min(X) \ll \#X$, что, в конечном счёте, и обеспечивает практическую эффективность предлагаемого алгоритма.

Фактически, наш алгоритм использует процедуру слияния двух отфильтрованных множеств $X = \Phi(X)$, $Y = \Phi(Y)$, в одно – $Z = \Phi(X \cup Y)$. Обращение к соответствующей процедуре мы будем записывать в виде

$$Z \leftarrow \mathrm{FILTR}(X, Y).$$

Для реализации процедуры FILTR можно использовать алгоритм из [4], трудоёмкость которого оценивается величиной $O\left((\#X + \#Y) log^{d-2}(\#X + \#Y)\right)$ при $d \geq 2$.

Так же, как и классические алгоритмы поиска оптимальных путей в графе [3], предлагаемый алгоритм решения задачи (2) относится к классу алгоритмов обхода графа, называемого *направленным поиском в ширину* (*directed* BFS-*algorithms*). В алгоритмах этого класса задача поиска решается в два этапа: на первом этапе для каждой вершины $v \in V$, достижимой из вершины s**,** находятся оценки допустимых путей из вершины s в данную вершину, а на втором этапе, называемом иногда *методом последовательного возвращения*, строится сам искомый путь.

II. Алгоритм

Вектор $p(\mu) = \left(w(\mu), \tau_1(\mu), \ldots, \tau_N(\mu)\right)$ мы называем *оценкой* пути $\mu \in \mathcal{M}(s, v)$. Поскольку в алгоритме используются только отфильтрованные множества оценок, то фиксация предпоследней вершины в каждом таком пути устанавливает взаимно – однозначное соответствие между множеством путей и отфильтрованным множеством их оценок, что позволяет привязать оценки к вершинам, а не к путям. Отфильтрованную совокупность оценок вершины $v \in V$ мы будем обозначать $P(v)$, а запись $v = pred(u|p)$, где $p \in P(u)$, означает, что из множества $\{\mu \in \mathcal{M}(s, u) : p(\mu) = p\}$ выбран путь, предпоследняя вершина которого есть v.

При описании алгоритма мы так же используем следующие обозначения: $\Gamma^+(v)$ – множество вершин, непосредственно следующих за вершиной v в графе G; Q - очередь, из которой выбираются вершины для корректировки совокупности оценок; u – текущая вершина; $track$ – переменная, в которой формируется искомый путь; α – фиктивная вершина. Из вершины α проводится дуга в вершину s, которая снабжается нулевыми весами. Начальные значения переменных задаются следующим образом: $P(s) = (0, \ldots 0)$, $pred\left(s|(0, \ldots, 0)\right) = \alpha$; $P(v) = \emptyset$ для $v \in V \backslash \{s\}$; $Q = \{s\}$; $track = \emptyset$.

Этап 1.

пока $Q \neq \emptyset$,

$u \leftarrow \mathrm{DEQUEUE}(Q)$, $P \leftarrow P(u)$;

для *каждой вершины* $v \in \Gamma^+(u)$

$Y \leftarrow \emptyset, X \leftarrow P(v)$; $a = (u, v)$,

$\tau \leftarrow \left(\tau_1(a), \ldots, \tau_2(a)\right)$;

для *каждой оценки* $p \in P$

если $\tau + (p_2, \ldots p_{N+1}) \leq (T_1, \ldots, T_N)$

$y \leftarrow (w(a), \tau) + p$; $Y \leftarrow Y \cup \{y\}$;

$pred(v|y) \leftarrow u$;

$Z \leftarrow \mathrm{FILTR}(X, Y)$;

если $Z \neq X$

$Q \leftarrow \mathrm{ENQUEQE}(Q, v)$;

если $P(t) = \emptyset$ ***вернуть*** track; ***останов***

иначе

*Этап 2***.**

$p_* \leftarrow argmin\{w(p) |\, p \in P(t)\}$; $u \leftarrow t$; $track \leftarrow t$; $p \leftarrow p_*$; $v \leftarrow pred(t|p)$; $a = (v, u)$;

пока $v \neq \alpha$

$track \leftarrow (v, track)$; $p \leftarrow p - \left(w(a), \tau_1(a), \ldots \tau_N(a)\right)$;

$u \leftarrow v$; $v \leftarrow pred(u|p)$; $a = (v, u)$;

вернуть track***;*** ***останов***.

III. Анализ алгоритма

Теорема 1. 1) Если для некоторого из весов $\tau_1, \dots, \tau_N$ все циклы в графе G имеют положительный вес, то алгоритм останавливается через конечное число шагов.

2) Если по завершении работы алгоритма $\mathrm{track} = \emptyset$, то задача (2) не имеет решения.

3) Если задача (2) имеет решение, то при завершении алгоритма переменная $track$ содержит последовательность вершин, задающих путь, который является одним из решений задачи (2).

Обозначим через d трудоёмкость алгоритма, т.е. суммарное количество элементарных операций (сравнений, суммирований и присваиваний), которые выполняются алгоритмом за время его работы. Пусть L_{max} – наибольшая из длин путей, удовлетворяющих ограничениям задачи (2), $\Delta_0 = max\{|w(a)|: a \in A\}$, $\Delta_i = max\{|\tau_i|: a \in A\}$, $i = 1, \dots N$; $\Delta = \sqrt[N+1]{\Delta_0 \Delta_1 \dots \Delta_N}$, $n = \#V$.

Теорема 2. *Для любого* $\varepsilon > 0$

$$d = O(n^2 (L_{max}\Delta)^{2N+1+\varepsilon})$$

Следствие 1. *Предположим, что все веса в задаче* (2) *ограничены константами, независящими от* n *и выполнены условия теоремы 1. Тогда для любого* $\delta > 0$

$$d = O\left(n^{2+\delta}\right).$$

Следствие 2. *Пусть n фиксировано,* $N = 1, \tau > 0, T > 0$. *Тогда для любого* $\varepsilon > 0$

$$d = O\left((\Delta_0 T)^{(3+\varepsilon)/2}\right).$$

На рис. 1 приведён граф, на котором иллюстрируется результат работы алгоритма в случае $N = 1, \tau > 0$. Первое число на дуге – w-вес, второе – τ-вес. Рис. 2 иллюстрирует чувствительность решений задачи (1) к варьированию порогового параметра T.

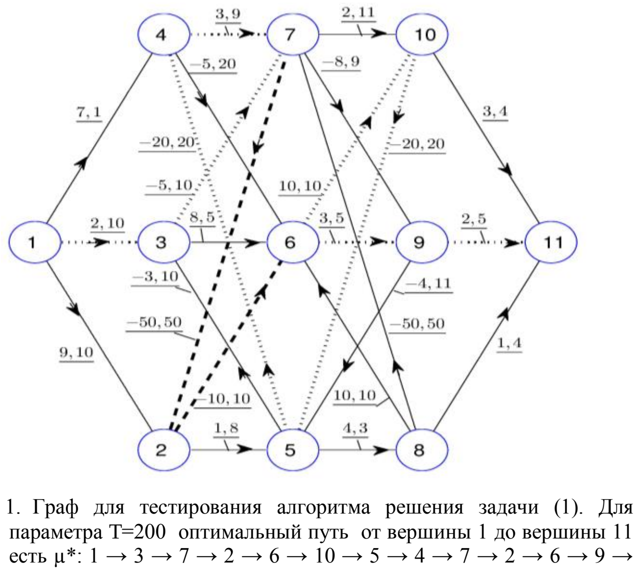

Рис. 1. Граф для тестирования алгоритма решения задачи (1). Для параметра T=200 оптимальный путь от вершины 1 до вершины 11 есть μ*: 1 → 3 → 7 → 2 → 6 → 10 → 5 → 4 → 7 → 2 → 6 → 9 → 11. Оптимальные значения весов есть w(μ*)=-145, τ(μ*))=199

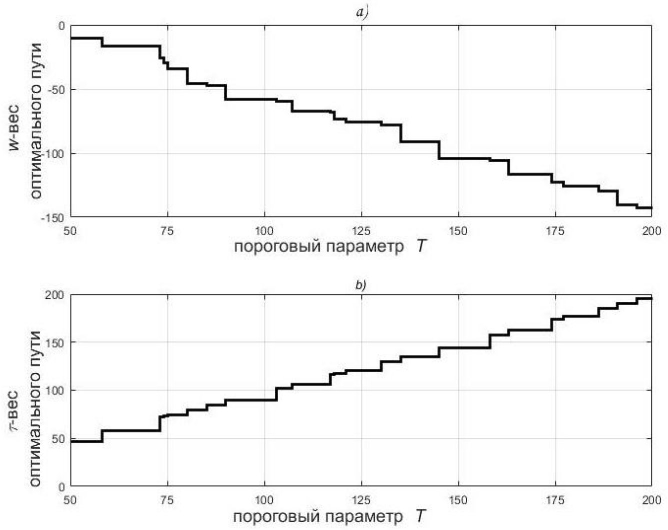

Рис. 2. Зависимость w-веса (a) и τ-веса (b) оптимального пути от величины порога T для графа, изображенного на рис. 1

Список литературы

[1] Moore E. F. The shortest path through a maze // Proc. Internat. Sympos. Switching Theory 1957, Part II. Cambridge, Mass.: Harvard Univ. Press., pp. 285–292.

[2] Панкратьев Е.В., Чеповский А.М., Черепанов Е.А., Чернышев С.В. Алгоритмы и методы решения задач составления расписаний и других экстремальных задач на графах большой размерности // Фундаментальная и прикладная математика, 2003, т.9, в.1, с. 235-251.

[3] Кормен Т, Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и анализ. М.:" И.Д. Вильямс", 2013. 1328 с. ISBN 978-5-8459-1794-2.

[4] Препарата Ф., Шеймос М.. Вычислительная геометрия. Введение. М: Мир, 1989. 478 с.