ОБОЗРЕНИЕ
ПРИКЛАДНОЙ И ПРОМЫШЛЕННОЙ
Том 20 **МАТЕМАТИКИ** **Выпуск 4**
2013

А. Н. Баушев, О. Л. Семёнова (СанктПетербург, ПГУПС, СПбГУ).
Об алгоритме поиска оптимального пути в ориентированном графе при аддитивных ограничениях.

Пусть $G = \langle V, A\rangle$ — конечный ориентированный граф на множестве вершин V с множеством дуг A; $w, \tau_1, \ldots, \tau_N : A \to R$ — набор весовых функций на множестве дуг графа G; $T = (T_1, \ldots, T_N)$ — набор вещественных чисел; $s, t \in V$ — две выделенные вершины в графе G.

Пусть для любого пути в G его w- и τ_i — веса определяются как суммы соответствующих весов входящих в него дуг. Обозначим через $M^T_{s,t}$ множество всех путей μ в графе G с начальной вершиной s и конечной вершиной t, удовлетворяющих ограничениям

$$\tau_i(\mu) \leqslant T_i, \quad i \in \{1, \ldots, N\} \tag{1}$$

и рассмотрим оптимизационную задачу

$$w(\mu) \to \min, \quad \mu \in M^T_{s,t}. \tag{2}$$

Задача (2) изучалась в [1], где для случая $N = 1$, неотрицательной функции w и положительной функции τ_1 был предложен алгоритм ее решения.

Предлагаемый нами алгоритм решения задачи (2) корректно работает в случаях, когда и w-вес и τ_i-веса могут иметь произвольные знаки. Достаточным условием корректной работы алгоритма является следующее условие допустимости весов.

Набор весов $w, \tau_1, \ldots, \tau_N$ называется *допустимым*, если

1) веса $w, \tau_1, \ldots, \tau_N$ являются целочисленными;

2) хотя бы в одной из сетей $G_i = \langle V, A, \tau_i\rangle$, $i \in \{1, \ldots, N\}$, отсутствуют циклы как отрицательного, так и нулевого веса.

Детальный анализ алгоритма и полиномиальная оценка времени его работы приведены в [3]. Здесь мы ограничимся его описанием,дающим возможность читателю реализовать алгоритм.

Путь μ в графе G можно задать последовательностью вершин, образующих его дуги. Такое представление пути мы будем называть его *треком* на множестве V и использовать запись:

$$\mu : v_1 \to v_2 \to \cdots \to v_{i-1} \to v_i \to \cdots \to v_{k-1} \to v_k.$$

При этом пути с треками вида $\mu_0 : v_1 \to v_2 \to \cdots \to v_{i-1} \to v_i$ называются начальными отрезками пути μ.

Для произвольной вершины $v \in V$ символом $M_{s,v}$ обозначим совокупность всех путей в графе G от вершины s до вершины v, которые являются начальными отрезками хотя бы для одного пути из множества $M^T_{s,t}$.

Вектор $p(\mu) = (w(\mu), \tau_1(\mu), \ldots, \tau_N(\mu))$ будем называть *оценкой* пути μ.

Функция $\theta : V \to R^N$ называется *пороговой функцией* для задачи (2), если для любой вершины $v \in V$ и для любого пути $\mu \in M_{s,v}$ выполняется неравенство $\tau(\eta) \leqslant \theta(v)$, где $\tau(\mu) = (\tau_1(\mu), \ldots, \tau_N(\mu))$.

Если все τ_i-веса неотрицательны, то, очевидно, функция $\theta(v) \equiv T$ является пороговой функцией для задачи (2). Для случая τ_i-весов произвольных знаков один из вариантов определения пороговой функции приведен в [3].

Пусть $(P, \leqslant)$ — частично упорядоченное множество, $P_1 \subset P$. Элемент $x \in P_1$ называется *минимальным* элементом P_1, если $\{y \in P_1 \,|\, y < x\} = \varnothing$. Множество всех минимальных элементов множества P_1 называется множеством Парето для множества P_1 по отношению к частичному порядку $\leqslant$. Отображение F, которое каждому подмножеству множества P ставит в соответствие его множество Парето, мы будем называть *фильтрацией* совокупности всех подмножеств отношением $\leqslant$, а неподвижные точки этого отображения отфильтрованными множествами.

Алгоритмы построения множества Парето хорошо известны.Фактически нам понадобится процедура фильтрации объединения двух отфильтрованных множеств оценок по нескольким последним координатам. Кроме того, в алгоритме используется признак того, что результат фильтрации объединения отличается от первого из объединяемых множеств. Обращение к соответствующей процедуре мы будем записывать в виде $[FP, mark] \leftarrow FILTRATION\_UNION(P_1, P_2, k)$. Здесь P_1, P_2 — отфильтрованные множества оценок, k — номер компоненты векторов оценок, начиная с которого осуществляется фильтрация, FP — результат фильтрации объединения $P = P_1 \cup P_2$, а выходной аргумент $mark = 1$, если $FP \neq P_1$ и $mark = 0$ в противном случае.

Предлагаемый нами алгоритм решения задачи (2) по существу представляет собой адаптацию классического алгоритма Беллмана–Мура к ограничениям задачи (2).

Алгоритм состоит из двух этапов. На первом этапе ищется оценка пути, являющегося решением (2). В процессе первого этапа алгоритма формируется очередь, состоящая из вершин, которые необходимо просмотреть в качестве текущей вершины для расширения и фильтрации построенной до этого совокупности оценок вершин, непосредственно следующих за текущей вершиной. Очередь мы будем представлять одномерным массивом Q длины $\#V$, в котором $Q(v) = 1$, если вершина v находится в очереди, и $Q(v) = 0$, иначе. В процессе алгоритма одна и та же вершина может несколько раз включаться в очередь и покидать ее, когда эта вершина выбирается в качестве текущей. Текущую вершину мы будем обозначать u.

Для решения задачи построения оптимального пути по найденной совокупности оценок (второй этап алгоритма) мы на первом этапе будем формировать для каждой вершины, достижимой из вершины s, множество *расширенных* оценок допустимых путей. В качестве расширенной оценки пути $\mu \in M_{s,v}$ мы рассматриваем вектор $\widetilde{p}(\mu) = (r, w(\mu), \tau_1(\mu), \ldots, \tau_N(\mu))$, где $r \in \Gamma_-(v)$ — вершина, непосредственно предшествующая вершине v в пути μ. Совокупность расширенных оценок начальных отрезков допустимых путей с конечной вершиной v обозначается $P\{v\}$. Множество всех вершин, непосредственно следующих за вершиной v, обозначается $\Gamma_+(v)$.

Будем считать, что вершины графа G занумерованы натуральными числами $1, \ldots, \#V$ и дополним граф фиктивной вершиной -1, символизирующей начало алгоритма. Из этой вершины выходит единственная дуга в вершину s с нулевыми значениями w- и τ-весов.

Алгоритм BS.

Вход: граф G с w- и τ- весами дуг, начальная и конечная вершины искомого пути s и t, пороговые значения для τ-весов $\theta(v)$, $v \in V$.

Выход: трек пути, который является решением (2).

Используемые функции: FILTRATION _ UNION

Первый этап

Шаг 1. Положить $Q(s) = 1$, $Q(v) = 0$, $v \in V\backslash\{s\}$, $P\{s\} = (-1, 0, \ldots, 0) \in \mathbf{R}^{N+2}$, $P\{v\} = \varnothing$, $v \in V\backslash\{s\}$, $q = 1$, перейти к шагу 2.

Шаг 2. Если $q = 0$ перейти к шагу 12, иначе перейти к шагу 3.

Шаг 3. Выбрать вершину $u : Q(u) = 1$, положить $Q(u) = 0$, $q = q - 1$, перейти к шагу 4.

Шаг 4. Положить $Z = \Gamma_+(u)$, $Pu = P\{u\}$, перейти к шагу 5.

Шаг 5. Если $\mathrm{Z} \neq \varnothing$, выбрать $z \in Z$, положить $Z = Z\backslash\{z\}$, $X = \varnothing$, перейти к шагу 6, иначе перейти к шагу 2.

Шаг 6. Если $Pu \neq \varnothing$, выбрать $p \in Pu$, положить $Pu = Pu\backslash\{p\}$, перейти к шагу 7, иначе перейти к шагу 8.

Шаг 7. Если $p_3 + \tau_1((u,z)), \ldots, p_{N+2} + \tau_N((u,z))) \leqslant \theta(z)$ положить $X = X \cup \{(u, p_2 + w((u,z)), p_3 + \tau_1((u,z)), \ldots, p_{N+2} + \tau_N((u,z)))\}$, перейти к шагу 6,иначе перейти к шагу 6.

Шаг 8. Положить $[Y, mark] \leftarrow FILTRATION\_UNION(P\{z\}, X, 2)$, перейти к шагу 9.

Шаг 9. Если $mark = 0$, перейти к шагу 5, иначе перейти к шагу 10.

Шаг 10. Положить $P\{z\} = Y$, перейти к шагу 11.

Шаг 11. Если $Q(z) = 1$, перейти к шагу 5, иначе положить $Q(z) = 1$, $q = q + 1$, перейти к шагу 5.

Второй этап

Шаг 12. Если $P\{t\} = \varnothing$, останов, иначе положить $track = t$, найти $p^* = \arg\min\{p_2 | p \in P\{t\}\}$, положить $u = p_1^*$, $weight = (p_2^*, \ldots, p_{N+2}^*)$, перейти к шагу 13.

Шаг 13. Если $u = -1$, останов, иначе положить $track = u \rightarrow track$, перейти к шагу 14.

Шаг 14. Найти $p \in P\{u\}$, для которого выполняется равенство $(p_2, \ldots, p_{N+2}) + (w(p_1, u), \tau_1(p_1, u), \ldots, \tau_N(p_1, u)) = weight$, положить $u = p_1$, $weight = (p_2, \ldots, p_{N+2}$ перейти к шагу 13.

Эффективность алгоритма BS [3], т. е. существование полиномиальной оценки зависимости времени его работы от параметров задачи, обусловлена тем обстоятельством, что, несмотря на возможный экспоненциальный рост числа путей, удовлетворяющих ограничениям задачи (2), совокупность используемых оценок этих путей имеет полиномиальный порядок роста. Использование процедуры фильтрации в типичных ситуациях значительно сокращает время поиска оптимального пути (и используемую память), так как математическое ожидание числа элементов множества Парето для случайного множества точек существенно меньше, чем число элементов самого множества.

На рис. 1 показан граф, который использовался для тестирования алгоритма BS в случае $N = 1$ и положительной функции τ_1. На рис. 2 показаны графики зависимости w- и τ-весов оптимального пути от параметра $T = T_1$.

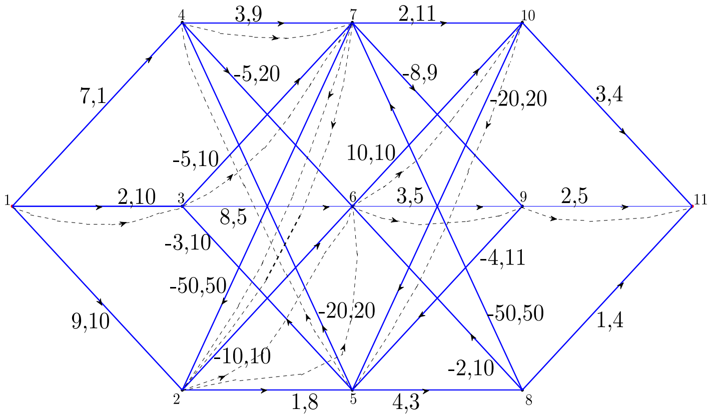

Рис. 1. Сеть для тестирования алгоритма BS. Первое число на дуге w-вес, второе — τ-вес. Пунктирными линиями показан путь μ : $1 \to 3 \to 7 \to 2 \to 6 \to 10 \to 5 \to 4 \to 7 \to 2 \to 6 \to 9 \to 11$, который является решением задачи (2), в которой $s = 1$, $t = 11$, $T = 200$, $w(\mu) = -145$, $\tau(\mu) = 199$

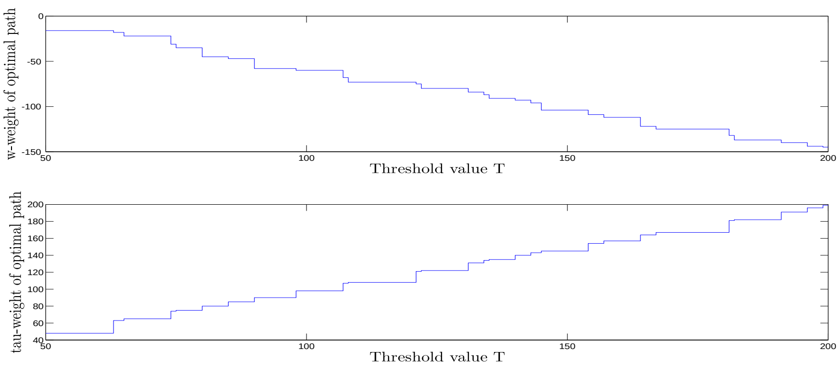

Рис. 2. Графики зависимостей w-веса и τ-веса оптимального пути от параметра T.

СПИСОК ЛИТЕРАТУРЫ

1. *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 больших размерностей. — Фундаментальная и прикладная математика, 2003, т. 9, в. 1, с. 235–251.
2. *Moore E. F.* The shortest path through a maze. Proc. Internat. Sympos. Switching Theory 1957, Part II. Cambridge, Mass.: Harvard Univ. Press., p. 285–292.
3. *Baushev A. N.*, *Semenova O. L.* Searching for optimal paths in graphs under additive constraints. — J. Graph Algorithms and Applications (отправлена в редакцию).