K. Kaymakov, D. S. Malyshev, Effective calculation of all tolerances in the sparse maximin routing problem, *Uspekhi Mat. Nauk*, 2024, Volume 79, Issue 5, 185–186

DOI: 10.4213/rm10199

Эффективное вычисление всех допусков в разреженной задаче о максиминном пути

К. В. Каймаков, Д. С. Малышев

Допуски элементов задач комбинаторной оптимизации (ЗКО) относительно их оптимальных решений дают информацию об устойчивости этих решений относительно изменений стоимостей элементов, а также служат инструментом для построения решателей ЗКО (см., например, [1], [2]). В [3] для задачи о максиминном пути и графа с n вершинами и m ребрами предложен алгоритм сложности $O(m+n\log n)$ вычисления оптимального решения и допусков всех m ребер относительно него. В этой работе мы предлагаем алгоритм сложности $O(m\alpha(m,n))$, вычисляющий те же параметры, где $\alpha(\,\cdot\,,\cdot\,)$ – обратная функция Аккермана; это асимптотически улучшает результат из [3] для разреженных (например, когда $m=O(n)$) графов.

Задача о максиминном пути (ЗМП) для заданных связного обыкновенного графа $G=(V_G,E_G)$, пропускных способностей ребер $c\colon E_G\to\mathbb{R}_{\geqslant 0}$ и вершин $s,t\in V_G$ состоит в том, чтобы найти $\arg\max_{P\in\mathcal{P}_{st}}\ \min_{e\in P}c(e)$, где $\mathcal{P}_{st}$ – множество всех путей между s и t. Эта задача возникает в качестве подзадачи при вычислении максимального потока в сети (см., например, [4] и [5]). Считаем, что $c(e_1)\neq c(e_2)$, если $e_1\neq e_2$.

Пусть P^* – произвольный максиминный st-путь графа (G,c), $\arg\min_{e\in P^*}c(e)=e^*$, а e – произвольное ребро G. *Верхним допуском* $u_{P^*}(e)$ (*нижним допуском* $l_{P^*}(e)$) *ребра e относительно* P^* называется супремум тех чисел $\alpha\geqslant 0$, для которых P^* остается максиминным st-путем в графе $(G,c_{+\alpha})$ (в графе $(G,c_{-\alpha})$), где $c_{\pm\alpha}(e')=c(e')$ для любого $e'\neq e$, а $c_{\pm\alpha}(e)=c(e)\pm\alpha$.

Наш алгоритм вычисления всех допусков для ЗМП основан на известной связи оптимальных решений ЗМП и задачи о максимальном остовном дереве. *Задача о максимальном остовном дереве* (ЗМОД) для заданного взвешенного по ребрам графа состоит в том, чтобы найти в нем поддерево, содержащее все вершины графа, с максимальной суммой весов ребер. Хорошо известно (см., например, [6]), что для любых $x,y\in V_G$ путь $T(x,y)$ между x и y в $T=(V_T,E_T)$ – произвольном МОД графа (G,c) является его максиминным xy-путем. Это приводит к следующей стратегии вычисления допусков для ЗМП: отслеживать изменения МОД и минимумов в st-путях при изменении пропускных способностей отдельных ребер. Рассмотрим ребра

$$\forall\, xy\in E_G\setminus E_T\quad \arg\min_{\{x'y'\colon x'y'\in T(x,y)\}}c(x'y'),\qquad \forall\, xy\in E_T\quad \arg\max_{\{x'y'\colon xy\in T(x',y')\}}c(x'y'),\tag{1}$$

которые назовем *заменяющими*. Если e' – заменяющее ребро для e, то деревья $T'=(T\setminus\{e'\})\cup\{e\}$ (если $e\in E_G\setminus E_T$) и $T'=(T\setminus\{e\})\cup\{e'\}$ (если $e\in E_T$) являются МОД графа (G,c) среди его остовных деревьев, содержащих $e\in E_G\setminus E_T$ или не содержащих $e\in E_T$ соответственно. Для графа (G,c) вычисление оптимального решения ЗМОД и всех заменяющих ребер относительно него выполняется [7] за время $O(m\alpha(m,n))$.

Ясно, что $u_{P^*}(e)=+\infty$ для любого $e\in E_T$ и $l_{P^*}(e)=+\infty$ для любого $e\in E_G\setminus E_T$, так как соответствующие изменения $c(e)$ не меняют T как МОД. По той же причине, если допуск e конечен, то существует заменяющее ребро e' и этот допуск не меньше $|c(e')-c(e)|$. Из определения допусков следует, что если $u_{P^*}(e)<+\infty$, то $u_{P^*}(e)\leqslant c(e^*)-c(e)$. Предположим, что $e\in E_G\setminus E_T$. Тогда существует заменяющее ребро e'. Если $e'\notin T(s,t)$, то $u_{P^*}(e)=+\infty$, так как $T(s,t)=T'(s,t)$ при любых увеличениях $c(e)$. Если же $e'\in T(s,t)$, то $u_{P^*}(e)=c(e^*)-c(e)$ при $c(e')=c(e)$

Работа поддержана грантом для научных центров в области искусственного интеллекта, предоставленным Аналитическим центром при Правительстве РФ в соответствии с соглашением о субсидировании (номер соглашения 000000D730324P540002) и соглашением с Московским физико-техническим институтом от 1 ноября 2021 г. № 70-2021-00138.

DOI: https://doi.org/10.4213/rm10199

и $u_{P^*}(e) = +\infty$ в противном случае (очевидно, что $c(e') \geqslant c(e^*)$, причем $c(e') > c(e^*)$ тогда и только тогда, когда $e^* \in T(s,t)$). Если $e \in E_T$, то либо $l_{P^*}(e) = c(e) - c(e')$ (если существует заменяющее ребро e' и $e \in T(s,t)$), так как в силу рассуждений выше имеем $\min_{e \in T'(s,t)} c(e) \geqslant \min(c(e^*), c(e'))$, либо $l_{P^*}(e) = +\infty$.

Принадлежность произвольного ребра $e = xy \in E_G$ пути $T(s,t)$ распознается за время $O(1)$. Для этого выберем произвольную вершину $r \in V_T$ в качестве его корня. Тем самым возникает отношение "предок–потомок". Предполагается, что каждая вершина является потомком самой себя. *Наименьшим общим предком* вершин $x, y \in V_T$, обозначаемым через $\mathrm{LCA}(x,y)$, называется ближайшая к r вершина, для которой x и y одновременно являются потомками. В работе [8] был предложен алгоритм вычисления наименьшего общего предка любых двух вершин за время $O(1)$ при линейной по n (количеству вершин дерева) предобработке. Поиском в ширину от вершины r за время $O(n)$ вычислим глубины всех вершин в T, где *глубина* $|T(r,z)|$ *вершины* z обозначается через $d_T(z)$. Обозначим через T_x и T_y компоненты связности леса $T \setminus \{e\}$, которые содержат вершины x и y соответственно. Можно считать, что $d_T(y) = d_T(x) + 1$, тогда $r \in T_x$. Ясно, что $xy \in T(s,t)$ тогда и только тогда, когда s и t лежат в разных компонентах леса $T \setminus \{e\}$. Проверка того, что вершина $z \in \{s,t\}$ принадлежит T_y, состоит в проверке равенства $\mathrm{LCA}(z,y) = y$. В точности одна из вершин s и t должна обладать этим свойством.

Тем самым, наш алгоритм можно описать следующим образом.

(1) Вычислить максиминный путь P^* и ребро e^*.

(2) Вычислить T – МОД графа (G,c) и все заменяющие ребра относительно T.

(3) Выполнить LCA-предобработку дерева T, вычислить глубины всех его вершин.

(4) В цикле по $e \in E_G$: (4.1) если $e \in T(s,t)$, то $u_{P^*} = +\infty$, а $l_{P^*}(e) = c(e) - c(e')$ в случае существования заменяющего ребра e' и $l_{P^*}(e) = +\infty$ в противном случае; (4.2) если $e \notin T(s,t)$, то $l_{P^*} = +\infty$, а $u_{P^*}(e) = c(e') - c(e)$ в случае существования заменяющего ребра e' и $u_{P^*}(e) = +\infty$ в противном случае.

Совместное вычисление P^* и e^* выполняется за время $O(m)$ [6], [9], вычисление T и всех заменяющих ребер выполняется за время $O(m\alpha(m,n))$ [7], третий шаг выполняется за время $O(n)$, одна итерация цикла выполняется за время $O(1)$. Тем самым, общая сложность нашего алгоритма есть $O(m\alpha(m,n))$.

Список литературы

[1] M. Turkensteen, D. Malyshev, B. Goldengorin, P. M. Pardalos, *J. Global Optim.*, **68**:3 (2017), 601–622. **[2]** M. Turkensteen, G. Jäger, *Theoret. Comput. Sci.*, **937** (2022), 1–21. **[3]** R. Ramaswamy, J. B. Orlin, N. Chakravarty, *Math. Program.*, **102**:2 (A) (2005), 355–369. **[4]** J. Edmonds, R. M. Karp, *J. ACM*, **19**:2 (1972), 248–264. **[5]** G. Baier, E. Köhler, M. Skutella, *Algorithms–ESA* 2002, Lecture Notes in Comput. Sci., **2461**, Springer-Verlag, Berlin, 2002, 101–113. **[6]** K. V. Kaymakov, D. S. Malyshev, *Optim. Lett.*, **18**:5 (2024), 1273–1283. **[7]** B. Dixon, M. Rauch, R. E. Tarjan, *SIAM J. Comput.*, **21**:6 (1992), 1184–1192. **[8]** J. Fischer, V. Heun, *Combinatorial pattern matching*, Lecture Notes in Comput. Sci., **4009**, Springer-Verlag, Berlin, 2006, 36–48. **[9]** P. M. Camerini, *Inform. Process. Lett.*, **7**:1 (1978), 10–14.

К. В. Каймаков (K. V. Kaymakov)
Coleman Tech LLC
E-mail: kirill.kaymakov@mail.ru

Д. С. Малышев (D. S. Malyshev)
Национальный исследовательский университет
"Высшая школа экономики";
Московский физико-технический институт
(национальный исследовательский университет)
E-mail: dsmalyshev@rambler.ru

Представлено А. М. Райгородским
Принято редколлегией
15.08.2024