ДИСКРЕТНЫЙ АНАЛИЗ И ИССЛЕДОВАНИЕ ОПЕРАЦИЙ
Октябрь–декабрь 2021. Т. 28, № 4. С. 117–124

УДК 519.178 DOI 10.33048/daio.2021.28.710

WIN-WIN АЛГОРИТМ ДЛЯ ЗАДАЧИ $(k+1)$-LST/k-ПУТЕВОГО ПРЕДСТАВЛЕНИЯ

А. Г. Ключиков[1, a], *М. Н. Вялый*[1, 2, 3, b]

[1] Московский физико-технический институт (гос. университет), Институтский пер., 9, 141701 Долгопрудный, Россия

[2] Федеральный исследовательский центр «Информатика и управление» РАН, ул. Вавилова, 44, корп. 2, 119333 Москва, Россия

[3] Национальный исследовательский университет «Высшая школа экономики», Покровский б-р, 11, 109028 Москва, Россия

E-mail: [a] `klyuchikov@phystech.edu`, [b] `vyalyi@gmail.com`

Аннотация. Описывается win-win алгоритм, строящий за полиномиальное от $|V(G)|$ и k время для любого графа G и любого натурального числа k либо остовное дерево с хотя бы $k+1$ листьями, либо предоставляет путевое представление (path decomposition) ширины не большей k. Данный алгоритм оптимален в силу следствия теоремы о путевом представлении [1]. Библиогр. 5.

Ключевые слова: путевое представление, остовное дерево, задача k-LST, win-win алгоритм, параметрическая сложность.

1. Задача

Задача, решаемая в данной статье, берёт своё начало в теореме из статьи [1], связывающей остовное дерево и путевое представление.

Определение 1. *Путевым представлением* графа G называют последовательность подмножеств X_i вершин $V(G)$, обладающую следующими свойствами:

(1) для всякого ребра $(u,v) \in E(G)$ найдётся такое X_i, что обе вершины лежат в этом множестве: $u,v \in X_i$;

(2) для любых $i \leqslant j \leqslant k$ имеет место включение $X_i \cap X_k \subseteq X_j$.

Работа второго автора выполнена при поддержке Программы фундаментальных исследований НИУ ВШЭ, а также в рамках государственного задания ФИЦ ИУ РАН (проект № 0063–2019–0003).

Определение 2. *Путевой шириной* графа называют минимальную среди всех путевых представлений величину, на единицу меньшую размера максимального множества в путевом представлении:

$$\min_{(X_1,\dots,X_s)} \max_{X_i} |X_i| - 1.$$

Теорема 1 [1]. *Для всякого леса F любой граф с путевой шириной, не меньшей $|F| - 1$, содержит минор, изоморфный F.*

Из данной теоремы напрямую вытекает

Следствие 1. *Если G — связный граф и $k < |V(G)|$ натуральное, то*
(1) *либо путевая ширина графа не превосходит k,*
(2) *либо существует остовное дерево с как минимум $k+1$ листьями.*

Тем самым можно поставить следующую задачу.

Задача $(k{+}1)$-LST/k-путевого представления. *Для связного графа G и натурального числа k построить либо путевое представление графа G ширины не больше k, либо остовное дерево с как минимум $k+1$ листьями.*

Заметим, что задачи нахождения путевого представления графа ширины k и остовного дерева с большим чем k числом листьев NP-трудны [2, 3]. Другими словами, в предположении $\mathrm{P} \neq \mathrm{NP}$ не существует полиномиального по параметрам $n = |V(G)|$ и k алгоритма, который решал бы каждый пункт из следствия 1 по отдельности. Тем не менее в данном случае обе эти проблемы связаны единым теоретическим фактом (следствие 1), который допускает существование полиномиального по (n, k) алгоритма, который в ходе своей работы либо предоставит остовное дерево с большим числом листьев, либо достаточно узкое путевое представление. Такие алгоритмы называются *алгоритмами «выигрыш-выигрыш»* (win-win). Для краткости записи будем придерживаться английского названия.

Данная задача поднималась на IWPEC [4], где было отмечено без ссылок, что наилучший известный алгоритм за полиномиальное время выдаёт либо остовное дерево с по крайней мере $k+1$ листьями, либо путевое представление ширины не более $2k-1$.

2. Обозначения

Зафиксируем терминологию, которой будем придерживаться в данной статье.

Положим $N(X) = \{v \in G - X \mid \exists\, u \in X\colon\ (u, v) \in E(G)\}$, т. е. $N(X)$ — множество внешних соседей вершин из множества X. В случае, когда

нас будет интересовать множество соседей среди вершин подграфа $G-X$ для $Y \subseteq X$, будем писать $N_X(Y)$. Таким образом, $N(X) = N_X(X)$.

Будем называть вершины в X, имеющие соседа в множестве $G - X$, *границей множества* $X \subseteq V(G)$ и обозначать ∂X (другое возможное обозначение — $\mathrm{att}(X)$).

Множество листьев дерева T будем обозначать $L(T)$ или L. Так, для краткости, дерево T_i имеет множество листьев L_i.

Будем обозначать через l лист дерева с *наибольшим числом внешних соседей* среди всех листьев дерева T, т. е. такой лист, для которого максимален размер множества $N_X(l)$ смежных с l вершин в множестве $G - T$.

3. Win-win алгоритм

3.1. Описание алгоритма. Здесь и далее будем полагать, что граф связный. В противном случае определяем поиском в глубину компоненты связности и сводим задачу к рассмотренной.

Алгоритм строит последовательность деревьев $T_1 \subset \cdots \subset T_s$. В качестве начального дерева T_1 выбирается произвольная вершина графа G.

Алгоритм 1. Win-win алгоритм для задачи $(k{+}1)$-LST/k-путевого представления

Вход: G, k
Выход: Один из вариантов:
- T — остовное дерево с не менее чем $k+1$ листьями
- $(X_1, X_2, \ldots, X_s)$ — путевое представление ширины не больше k

$T := \{v\}$, $L := \{v\}$, $l = v$, $X_1 := v$, $i = 2$ $\quad\triangleright$ v — произвольная вершина G
while $V(T) \neq V(G)$ **do**
    $l := \operatorname*{argmax}_{v \in L} |N_T(v)|$
    $T' := T$
    $L' := L - \{l\} + N_T(l)$
    for $v \in N_T(l)$ **do**
        $T' := T + (l, v)$
    end for
    $X_i := \partial V(T) \cup V(T' - T)$
    $T := T'$; $L = L'$; $i := i + 1$;
end while
if $|L| \geqslant k + 1$ **then**
    вывести T
else
    вывести $(X_1, X_2, \ldots, X_i)$
end if

Пусть построено дерево T_k с множеством листьев $L_k \subset V(T_k)$. Выберем в этом множестве лист l_k с максимальным числом соседей в $G-T_k$. Дерево T_{k+1} получается присоединением $N_{T_k}(l_k)$ к l_k в дереве T_k. Процедура продолжается до тех пор, пока T_s не покроет все вершины графа G.

Если $|L_s| > k$, то дерево T_s — искомый остов с как минимум $k+1$ листьями. Иначе последовательность $(X_1, X_2, \ldots, X_s)$, где $X_i = \partial V(T_{i-1}) \cup V(T_i - T_{i-1})$, $X_1 = V(T_1)$, является искомым путевым представлением ширины не более k.

3.2. Время работы. Алгоритм представляет собой жадный обход вершин графа G, в процессе которого строится остовное дерево T и путевое представление $(X_1, \ldots, X_s)$ графа G.

Время работы алгоритма полностью определяется циклом **while**. Выделим самые ресурсоёмкие действия при условии, что арифметические операции выполняются за $O(1)$. На каждой итерации требуется найти вершину l — элемент множества с наибольшим числом внешних соседей:

$$l := \operatorname*{argmax}_{v \in L} |N_T(v)|.$$

Время работы этой операции зависит от конкретной реализации этой строчки. Тривиальная верхняя оценка даёт $O(|V(G)|^2)$. Её можно при желании улучшить до $O(|E(G)|)$ в среднем, если работать со списками смежности и хэш-таблицами.

Аналогичные оценки получаются для следующей строки алгоритма:

$$X_i := \partial V(T) \cup V(T' - T).$$

Поскольку на каждой итерации алгоритма в дерево T добавляется как минимум одна вершина, итоговая верхняя оценка равна $O(|V(G)|^3)$ или $O(|V(G)| \cdot |E(G)|)$ в среднем.

Основное тело алгоритма никак не оперирует параметром k. В самом конце алгоритма значение k сравнивается с числом листьев полученного остовного дерева, что в предположении о «быстрых» арифметических операциях даст аддитивную добавку $O(1)$.

3.3. Корректность алгоритма. Докажем корректность данного алгоритма.

Если итоговое дерево T_s с числом вершин $|V(G)|$ содержит $k+1$ листьев и более, то искомое остовное дерево, очевидным образом, найдено. Далее полагаем, что у T_s не более k листьев. Это также верно и для всякого промежуточного дерева.

Лемма 1. *Любое дерево T_i в G обладает тем свойством, что $\partial V(T_i) = \partial L_i \subseteq L(T_i)$. Иными словами, для любого ребра $(u, v) \in E(G)$, $v \in G - T_i$, $u \in T_i$, вершина u является листом.*

Доказательство. Утверждение леммы верно, поскольку на каждом шаге добавляются все возможные внешние соседи «расширяющейся» вершины. Индукция завершает доказательство.

Лемма 2 [1, 5]**.** *Последовательность* $(X_1, X_2, \ldots, X_s)$ *является путевым представлением.*

Доказательство. Данная лемма аналогична лемме 2.1 в [1], а также приведена в виде формулы (2.1) в [5]. Мы приводим подробное доказательство, не опирающееся на конструкцию «блокады», описанную в статьях, указанных выше.

Поскольку суть алгоритма заключается в последовательном обходе всех вершин и добавлении их в остовное дерево, в дальнейшем под «посещёнными» на шаге i вершинами будут пониматься вершины графа G, которые были добавлены в промежуточный остов T_i на шаге i.

Проверим два пункта из определения путевого представления. Для этого перепишем X_i:

$$X_i = \partial V(T_{i-1}) \cup V(T_i - T_{i-1}) = \partial L_{i-1} \cup N_{T_{i-1}}(l_{i-1}).$$

Отсюда видно, что множество X_i состоит из некоторых листьев дерева T_i и той единственной «расширяющейся» вершины $l_{i-1} \in \partial L_{i-1}$, которая листом в T_i не является.

Докажем п. (2) определения путевого представления. Если посещённая на шаге i вершина v не входит в X_j, $i \leqslant j$, то $v \neq l_{j-1}$ и

- либо v не является листом в дереве T_{j-1} ($v \notin L_{j-1}$),
- либо v — это лист такой, что все инцидентные ему ребра соединяют другие вершины дерева ($N_{T_{j-1}}(v) = \varnothing$).

В обоих случаях вершина v не добавится в X_{j+1} по построению множества X_{j+1}. Следовательно, не существует вершины u такой, что для любых $i \leqslant j \leqslant k$ имели бы место соотношения $u \in X_i$, $u \in X_k$, $u \notin X_j$; это и есть п. (2) определения 1.

Докажем п. (1) определения путевого представления. Для любого ребра $e = (u, v) \in E$ определим минимальный индекс j, когда одна из концевых вершин попадает в дерево. Если $u, v \in X_j$, то искомое очевидно. Иначе $u \in X_j$ и $v \notin X_i$ для любого $i \leqslant j$. Из первого следует, что $u \in N(l_{j-1})$, а из второго — что вершина v не была посещена на шагах вплоть до j. Возможны два следующих случая.

(1) На шаге k вершина u выбирается как «расширяющаяся» l_{k-1}, а v всё ещё не добавлена в дерево. Тогда u и v входят в X_k, так как $v \in N_{T_{k-1}}(l_{k-1})$ и $u = l_{k-1}$.

(2) На шаге k вершина v добавляется в дерево T_k, а $u \neq l_i$ для всех $i \leqslant k$. Тогда $v \in N_{T_{k-1}}(l_{k-1})$ и $u \in \partial T_{k-1}$ (так как это лист и существует ребро в (u, v), $v \notin T_{k-1}$). Значит, $u, v \in X_k$.

В обоих случаях найдётся такое k, что $u, v \in X_k$, а значит, п. (1) определения 1 выполняется. Лемма доказана.

Лемма 3. *Ширина путевого представления* $(X_1, X_2, \ldots, X_s)$ *не превосходит* k.

Доказательство. Так как для любого i множество X_i состоит исключительно из листьев промежуточного дерева T_i и ровно одной внутренней вершины l_{i-1}, размер максимального множества в путевом представлении $(X_1, X_2, \ldots, X_s)$ не превосходит числа листьев в T_s плюс одна вершина l_{s-1}, т. е. $k+1$. Таким образом, по определению путевая ширина меньше или равна k.

Теорема 2. *Существует win-win алгоритм, который за полиномиальное по* $|V(g)|$ *и* k *время выдаёт либо остовное дерево с не менее чем* $k+1$ *листьями, либо путевое представление ширины не больше* k.

Доказательство следует из описания алгоритма и лемм 1–3.

ЛИТЕРАТУРА

1. **Bienstock D., Robertson N., Seymour P. D., Thomas R.** Quickly excluding a forest // J. Comb. Theory, Ser. B. 1991. Vol. 52. P. 274–283.
2. **Douglas R. J.** NP-completeness and degree restricted spanning trees // Discrete Math. 1992. Vol. 105, No. 1–3. P. 41–47.
3. **Kashiwabara T., Fujisawa T.** NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph // Proc. 1979 Int. Symp. Circuits and Systems (Tokyo, Japan, July 17–19, 1979). New York: IEEE, 1979. P. 657–660.
4. **Bodlaender H. L., Demaine E. D., Fellows M. R.** [et al.]. Open problems in parameterized and exact computation — IWPEC 2008. Technical Report UU–CS–2008–017. Utrecht: Utrecht Univ., 2008. Available at `http://www.cs.uu.nl/research/techreps/repo/CS-2008/2008-017.pdf` (accessed Aug. 3, 2021).
5. **Diestel R.** Graph minors I: A short proof of the path-width theorem // Comb. Prob. Comput. 1995. Vol. 4, No. 1. P. 27–30.

Ключиков Артём Геннадиевич
Вялый Михаил Николаевич

Статья поступила
12 апреля 2021 г.
После доработки —
2 августа 2021 г.
Принята к публикации
3 августа 2021 г.

DISKRETNYI ANALIZ I ISSLEDOVANIE OPERATSII
/DISCRETE ANALYSIS AND OPERATIONS RESEARCH/
October–December 2021. Vol. 28, No. 4. P. 117–124

UDC 519.178 DOI 10.33048/daio.2021.28.710

A WIN-WIN ALGORITHM FOR THE $(k+1)$-LST/k-PATHWIDTH PROBLEM

A. G. Klyuchikov[1, a] and *M. N. Vyalyi*[1, 2, 3, b]

[1] Moscow Institute of Physics and Technology,
9 Institutskii Lane, 141701 Dolgoprudnyi, Russia

[2] Federal Research Center "Computer Science and Control" RAS
44 Bld. 2 Vavilov Street, 119333 Moscow, Russia

[3] HSE University,
11 Pokrovskii Boulevard, 109028 Moscow, Russia

E-mail: [a] `klyuchikov@phystech.edu`, [b] `vyalyi@gmail.com`

Abstract. We describe a win-win algorithm that produces in polynomial of size of a graph G and given parameter k time either a spanning tree with al least $k+1$ leaves or a path decomposition with width at most k. This algorithm is optimal due to the path decomposition theorem [1]. Bibliogr. 5.

Keywords: path decomposition, spanning tree, k-LST problem, win-win algorithm, FPT.

REFERENCES

1. **D. Bienstock, N. Robertson, P. D. Seymour,** and **R. Thomas,** Quickly excluding a forest, *J. Comb. Theory, Ser. B.*, **52**, 274–283 (1991).
2. **R. J. Douglas,** NP-completeness and degree restricted spanning trees, *Discrete Math.* **105** (1–3), 41–47 (1992).
3. **T. Kashiwabara** and **T. Fujisawa,** NP-completeness of the problem of finding a minimum-clique-number interval graph containing a given graph as a subgraph, in *Proc. 1979 Int. Symp. Circuits and Systems, Tokyo, Japan, July 17–19, 1979* (IEEE, New York, 1979), pp. 657–660.

The work of the second author is supported by the HSE University Program for Basic Research and carried out within the framework of the State Assignment of FRC "Computer Science and Control" RAS (Project 0063–2019–0003).

English version: Journal of Applied and Industrial Mathematics **15** (4) (2021).

4. **H. L. Bodlaender, E. D. Demaine, M. R. Fellows** [et al.], Open problems in parameterized and exact computation — IWPEC 2008, *Technical Report UU–CS–2008–017*, (Utrecht Univ., Utrecht, 2008). Available at `http://www.cs.uu.nl/research/techreps/repo/CS-2008/2008-017.pdf` (accessed Aug. 3, 2021).
5. **R. Diestel,** Graph minors I: A short proof of the path-width theorem, *Comb. Prob. Comput.* **4** (1), 27–30 (1995).

Artyom G. Klyuchikov
Mikhail N. Vyalyi

Received April 12, 2021
Revised August 2, 2021
Accepted August 3, 2021