УДК 519.165

Об алгоритме Бёрклунда подсчета числа совершенных паросочетаний

М. Н. Вялый*

Вычислительный центр им. А. А. Дородницына ФИЦ ИУ РАН

Как известно, задача подсчёта числа совершенных паросочетаний в графе $\#P$-трудна [1]. Все известные алгоритмы её решения работают за экспоненциальное время. Остаётся открытым вопрос о том, насколько можно уменьшить показатель экспоненты.

Для двудольных графов давно известен алгоритм Райзера [2], который работает за время $O^*(2^{n/2})$. Здесь n — количество вершин в графе, а $O^*(\cdot)$ обозначает асимптотическую оценку с точностью до множителя, ограниченного полиномом от n. Этот алгоритм основан на вычислении перманента квадратной матрицы с помощью формулы включений и исключений.

Для общего случая многочлен, перечисляющий паросочетания, называется хаффнианом. В 2012 году Бёрклунд [3] предложил алгоритм подсчёта хаффниана (и, тем самым, совершенных паросочетаний в произвольном графе), время работы которого также оценивается $O^*(2^{n/2})$. Алгоритм основан на сводимости задачи подсчёта хаффниана к вычислению более общего многочлена, отвечающего задаче о покрытии, с коэффициентами в подходящей алгебре.

В этой работе мы излагаем алгоритм Бёрклунда как алгоритм вычисления коэффициента многочлена в нильпотентной алгебре. В оригинальной работе Бёрклунда фактически также использовалась нильпотентная алгебра. Однако соотношения в ней записывались в комбинаторных терминах и представляли индуктивные формулы для вычисления хаффниана. Вместе с тем, как показано в этой работе, приведенное в работе [3] простое комбинаторное описание алгоритма подсчёта числа совершенных паросочетаний допускает буквальный перевод на язык нильпотентной алгебры.

Нам представляется, что такое изложение алгоритма Бёрклунда представляет интерес. Во-первых, становится более прозрачной сама структура алгоритма. Во-вторых, предлагаемая явная алгебраическая структура вычислений позволяет надеяться на новые приложения этой идеи к задачам подсчёта совершенных паросочетаний в ограниченных классах графов.

Для простоты изложения мы говорим о подсчёте числа совершенных паросочетаний. Заметим однако, что тот же алгоритм можно применить и для вычисления хаффниана.

1 Алгебраическая формулировка задачи подсчёта совершенных паросочетаний

Пусть $G(V,E)$ — граф с n вершинами и m рёбрами.

*Работа выполнена при поддержке РФФИ, номер проекта 14–01–00641

Каждой вершине u графа G сопоставим переменную u, для удобства обозначаем её той же буквой. Каждому ребру $(uv) \in E(G)$ сопоставим двучлен

$$D_{uv} = 1 + uv.$$

Всему графу сопоставим произведение этих двучленов

$$D_G = \prod_{(uv)\in E(G)} D_u. \tag{1}$$

Это многочлен степени m от n переменных.

Для набора показателей $\alpha = (\alpha_1, \dots, \alpha_n)$ через $[\alpha]f$ обозначаем коэффициент в многочлене f при мономе $x^\alpha = \prod_i x_i^{\alpha_i}$.

Утверждение 1. *Количество M_G совершенных паросочетаний в графе G равно*

$$M_G = [1^n]D_G.$$

Доказательство. Раскрывая скобки в произведении (1), получаем представление D_G в виде суммы мономов. Каждый из этих мономов задаётся подмножеством рёбер $S \subseteq E(G)$, а набор показателей у монома в точности равен набору степеней в графе (V, S).

Осталось заметить, что графы со степенями всех вершин 1 — это в точности совершенные паросочетания. □

Аналогично можно выразить результат подсчёта в задаче точного покрытия. Пусть $\mathcal{F} = \{S_1, \dots, S_m\}$ — семейство подмножеств множества $[n]$. Точным покрытием назовём такое подсемейство $\mathcal{F}(I) = \{S_{i_1}, \dots, S_{i_r}\}$, где $I = \{i_1, \dots, i_r\}$, что

$$S_i \cap S_j = \varnothing, \quad \text{если } i, j \in I,$$
$$\bigcup_{i\in I} S_i = [n].$$

Через $x(S)$ обозначим произведение переменных x_i, $i \in S$.

Утверждение 2. *Количество точных покрытий равно коэффициенту $[1^n]D_{\mathcal{F}}$ в многочлене*

$$D_{\mathcal{F}} = \prod_{j\in[m]} (1 + x(S_j)).$$

Доказательство аналогично предыдущему.

2 Нильпотентные алгебры и отображения склейки

Нильпотентной алгеброй мы называем кольцо вычетов

$$R_n = \mathbb{R}[x_1, \dots, x_n]/I, \qquad I = (x_i^2 : i \in \{1, \dots, n\}).$$

В каждом классе вычетов есть ровно один полилинейный многочлен. Поэтому элементы R_n будем задавать полилинейными многочленами. В частности, говоря о коэффициентах элементов из R_n или о его степени, имеем в виду коэффициенты соответствующего полилинейного многочлена и его степень. Общее количество коэффициентов у полилинейных многочленов равно 2^n, это размерность нильпотентной алгебры.

Степенью одночлена по множеству переменных S называем сумму степеней переменных, входящих в S и в этот одночлен. Степень многочлена по множеству переменных S определяется как максимальная степень по одночленам, входящим в этот многочлен.

Если степень многочлена по некоторой переменной равна 0, говорим, что эта переменная не входит в многочлен.

Сложение в алгебре R_n выполняется за $O(2^n)$ операций, если элементы кольца представлены полилинейными многочленами. Это просто покомпонентное сложение векторов коэффициентов. Умножение в кольце R_n возможно за $O(n^2 2^n)$ операций. Для этого применяются ранжированные преобразования Мёбиуса и обратное (дзета-функция в алгебре инцидентности порядка на подмножествах) [4].

Из полилинейного представления классов вычетов сразу получается следующее утверждение

Утверждение 3. *Если $f - g \in I$, то*

$$[1^n]f = [1^n]g.$$

Поэтому вычисление коэффициента $[1^n]D_G$ можно проводить в нильпотентной алгебре.

Нам потребуется несколько простых фактов о нильпотентной алгебре.

Утверждение 4. *Пусть $f_1, \dots, f_k$ — многочлены, не содержащие переменную u. Тогда в R_n выполняется равенство*

$$\prod_{j=1}^{k}(1 + uf_j) = 1 + u\sum_{j=1}^{k} f_j.$$

Доказательство. Достаточно заметить, что $u^2 = 0$. □

Как следствие из утверждения 4 получаем для любого монома $x(S)$, $S \neq \varnothing$, равенство в кольце R_n

$$(1 + \lambda x(S))^d = 1 + d\lambda x(S). \tag{2}$$

Для изложения алгоритма Бёрклунда мы используем линейные отображения $R_n \to R_{n-1}$ «склейки» между нильпотентными алгебрами. Определим их так. Пусть u, v — две переменные из множества $\{x_1, \dots, x_n\}$. Добавим переменную ξ условием $\xi = uv$ и исключим u, v. Более формально, зададим φ_{uv} на базисе R_n, состоящем из полилинейных мономов:

$$\varphi_{uv}(f) = \begin{cases} \xi g, & \text{если } f = uvg, \text{ где } g \text{ не содержит } u,\ v, \\ 0, & \text{если } f = ug \text{ или } f = vg, \text{ где } g \text{ не содержит } u,\ v, \\ f & \text{в остальных случаях.} \end{cases} \tag{3}$$

Образ $\varphi_{uv}(f)$ лежит в алгебре $R_{n-1}[\xi, x_i(i \neq u, i \neq v)]$.

Утверждение 5. $[1^n]f = [1^{n-1}]\varphi_{uv}(f)$.

Доказательство. Достаточно проверить равенство для мономов. □

Утверждение 6 (слабая мультипликативность)**.** *Пусть $f, g \in R_n$, причём в g нет мономов, содержащих переменные u, v. Тогда*

$$\varphi_{uv}(fg) = \varphi_{uv}(f)\varphi_{uv}(g).$$

Доказательство. В силу линейности достаточно проверить это соотношение для произведений мономов. □

3 Алгоритм Бёрклунда в нильпотентной алгебре

Вычисление D_G путём раскрытия скобок и приведения подобных в кольце R_n требует времени $O^*(2^n)$.

Ускорить вычисление возможно, если применять отображения «склейки» (в силу утв. 5 коэффициент, отвечающий совершенным паросочетаниям, не изменяется при склейке).

Мы сведём вычисление коэффициента $[1^n]D_G$ к вычислению коэффициента $[1^{n/2}]F$ для некоторого многочлена от вдвое меньшего количества переменных. Этот многочлен будет задаваться алгебраической схемой размера $\mathrm{poly}(n)$ в алгебре $R_{n/2}$. Поскольку арифметические операции в нильпотентной алгебре выполняются быстро, как уже было указано, вычисление всех коэффициентов такого многочлена требует времени $O^*(2^{n/2})$.

Сводящий алгоритм строит последовательность многочленов $F_0 = D_G, \ldots, F_{n/2} = F$. Многочлен F_k принадлежит кольцу R_{n-k}, переменные в этом кольце обозначим $x_1, \ldots, x_{n-2k}, \xi_1, \ldots, \xi_k$. Переменные x_i отвечают вершинам графа, поэтому мы их также обозначаем как u, v, как это делалось раньше. Множество $\{x_1, \ldots, x_{n-2k}\}$ обозначаем V_k.

Каждый из многочленов F_k представляется в виде

$$F_k = (1 + f_\varnothing^{(k)}) \prod_{u,v \in V_k} (1 + uv f_{uv}^{(k)}), \qquad f_{uv}^{(k)}, f_\varnothing^{(k)} \in R_k[\xi_1, \ldots, \xi_k]. \tag{4}$$

Нижние индексы в $f_{uv}^{(k)}$ считаем неупорядоченной парой.

Для $D_G = F_0$ имеем по определению $f_\varnothing^{(0)} = 0$, $f_{uv}^{(0)}$ равны весам пар вершин в исходном графе (вес 1, если пара связана ребром, и 0 в противном случае; при вычислении хаффниана эти веса являются значениями переменных, входящих в хаффниан).

Следующий в последовательности многочлен F_{k+1} определяется как склейка очередной пары переменных $\varphi_{x_{n-2k-1}x_{n-2k}}(F_k)$ с новой переменной ξ_{k+1}, приведённая к виду (4).

Возможность такого приведения выражает основную идею алгоритма.

Лемма 1. *Если F_k имеет вид (4), то и $F_{k+1} = \varphi_{x_{n-2k-1}x_{n-2k}}(F_k)$ можно представить в виде (4).*

Доказательство. Обозначим для краткости $u = x_{n-2k-1}$, $v = x_{n-2k}$, $\varphi = \varphi_{x_{n-2k-1}x_{n-2k}}$. Выделим в произведении (4) множители, содержащие переменные u, v:

$$F_k = (1 + uv f_{uv}^{(k)}) \prod_{w \neq v} (1 + uw f_{uw}^{(k)}) \prod_{w \neq u} (1 + vw f_{vw}^{(k)}) \tilde{F}_k,$$

здесь в $\tilde{F}_k$ не входят переменные u, v. Поэтому $\varphi(\tilde{F}_k) = F_k$, а

$$\begin{aligned}
\varphi\Big((1 + uv f_{uv}^{(k)}) &\prod_{w \neq v} (1 + uw f_{uw}^{(k)}) \prod_{w \neq u} (1 + vw f_{vw}^{(k)})\Big) = \\
&= \varphi\Big(uv f_{uv}^{(k)} + \prod_{w \neq v} (1 + uw f_{uw}^{(k)}) \prod_{w \neq u} (1 + vw f_{vw}^{(k)})\Big) = \\
&= \varphi\Big(1 + uv f_{uv}^{(k)} + \sum_{w', w''} uvw'w'' f_{uw'}^{(k)} f_{vw''}^{(k)}\Big) = \\
&= 1 + \xi_{k+1} f_{uv}^{(k)} + \xi_{k+1} \sum_{w', w''} w'w'' f_{uw'}^{(k)} f_{vw''}^{(k)} = \\
&= (1 + \xi_{k+1} f_{uv}^{(k)}) \prod_{w', w''} (1 + \xi_{k+1} w'w'' f_{uw'}^{(k)} f_{vw''}^{(k)}).
\end{aligned}$$

В последнем равенстве использовано утв. 4, а в третьем опущены слагаемые, которые обращаются в 0 при действии φ.

Итак,

$$F_{k+1} = (1 + \xi_{k+1} f_{uv}^{(k)}) \prod_{w', w''} (1 + \xi_{k+1} w' w'' f_{uw'}^{(k)} f_{vw''}^{(k)}) \tilde{F}_k.$$

Применим ещё раз утв. 4 и сгруппируем биномы, которые содержат одинаковые пары переменных x_i, x_j. Получаем выражение

$$F_{k+1} = \\ = (1 + \xi_{k+1} f_{uv}^{(k)})(1 + f_{\varnothing}^{(k)}) \prod_{u', v' \in V_{k+1}} \left(1 + u'v'(f_{u'v'}^{(k)} + \xi_{k+1}(f_{u'u}^{(k)} f_{v'v}^{(k)} + f_{u'v}^{(k)} f_{v'u}^{(k)}))\right) \tag{5}$$

в виде (4). □

Корректность сводимости $[1^n]D_G = [1^{n/2}]F_{n/2}$ следует из утв. 5. Осталось оценить время работы. Как уже говорилось, эта оценка основана на оценке размера алгебраических схем, вычисляющих многочлены $f_*^{(k)}$.

Лемма 2. *Существует алгебраическая схема, вычисляющая все многочлены $f_*^{(*)}$ за $\mathrm{poly}(n)$.*

Доказательство. Многочлены $f_*^{(0)}$ (это константы) вычисляются схемой размера $O(n^2)$ (столько есть разных многочленов вида $f_*^{(0)}$).

Пусть многочлены $f_*^{(k)}$ вычисляются схемой размера N. Из (5) получаем рекуррентные соотношения

$$\begin{aligned} f_{\varnothing}^{(k+1)} &= (1 + \xi_{k+1} f_{uv}^{(k)})(1 + f_{\varnothing}^{(k)}), \\ f_{u'v'}^{(k+1)} &= f_{u'v'}^{(k)} + \xi_{k+1}(f_{u'u}^{(k)} f_{v'v}^{(k)} + f_{u'v}^{(k)} f_{v'u}^{(k)}) \end{aligned}$$

Для вычисления каждого из $O(n^2)$ многочленов, основываясь на предыдущих многочленах нужно $O(1)$ операций. Таким образом, общий размер схемы, вычисляющей все многочлены $f_*^{(*)}$ оценивается как $O(n^3)$. □

Из доказанных лемм и результатов [4] получаем

Теорема 1. *Существует алгоритм подсчёта числа совершенных паросочетаний в графе на n вершинах за время $O^*(2^{n/2})$.*

Литература

[1] *Valiant L.G.* The complexity of computing the permanent // Theoretical Computer Science. 1979. V. 8. P. 189–201.

[2] *Ryser H.J.* Combinatorial mathematics. New York–London-Sidney: John Wiley, 1963. 162 p.

[3] *Björklund A.* Counting perfect matchings as fast as Ryser // SODA'12 Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms. 2012. P. 914–921.

[4] *Björklund A., Husfeldt T., Kaski P., Koivisto M.* Fourier meets Möbius: Fast Subset Convolution. // Proceedings of the 39th ACM symposium on theory of computing (STOC). 2007. P. 67–74.