Стохастический поиск минимального покрытия булевой матрицы

A Preprint

С.М. Макеев
Факультет ВМК
МГУ им. М.В. Ломоносова
Москва
Sergey.Makeev@student.msu.ru

Е.В. Дюкова
ВЦ РАН
Москва
edjukova@mail.ru

Abstract

Задача о покрытии булевой матрицы, занимающая центральное место в дискретной математике, заключается в нахождении набора столбцов, удовлетворяющих определению покрытия, с наименьшим весом. Эта задача является одной из шести основных NP-полных задач, сложность тривиального алгоритма, известного как алгоритм дуализации, растет экспоненциально с ростом числа столбцов. Из-за вычислительных ограничений при больших размерах матриц не всегда удается решить задачу о покрытии точно. В настоящей работе проводится экспериментальное сравнение приближенных алгоритмов решения задачи о покрытии, заключающихся в поиске минимального покрытия среди достаточно представительной выборки неприводимых покрытий. Установлено, что для построения такой выборки оптимально использовать алгоритм RUNC-M, показывающий лучшие в плане вычислительной скорости результаты.

Keywords Покрытие · Жадный алгоритм · Стохастические алгоритмы · Асимптотически оптимальный алгоритм

1 Введение

+С ссылки

--- — везде по тексту

В настоящей работе рассматривается одна из центральных задач дискретной математики - задача о покрытии. Существующие алгоритмы полного перебора неприводимых покрытий позволяют решить ее с существенными ограничениями на размер булевой матрицы. При решении прикладных задач используются приближенные алгоритмы.

Важность задачи о покрытии обусловлена ее широким применением на практике. К ней сводится, например, задача о размещении пунктов обслуживания [8], она также используется в системах информационного поиска, при назначении экипажей на транспорте [8, 1], проектировании интегральных схем [7] и конвейерных линий и т.д.

Известен тривиальный алгоритм построения неприводимых покрытий булевой матрицы (алгоритм дуализации), заключающийся в перемножении дизъюнкций согласно дистрибутивному закону с последующим удалением из построенных конъюнкций повторяющихся переменных, а также повторяющихся и поглощенных конъюнкций. Время работы алгоритма очень быстро растет с ростом числа столбцов, оно увеличивается примерно в 2 раза при добавлении одного столбца. Таким образом, возникает вопрос о существовании более эффективных алгоритмов, который был поставлен более 40 лет назад.

Эффективность алгоритмов для перечислительных задач принято оценивать временем выполнения одного шага. Алгоритм с полиномиальной временной оценкой (алгоритм с полиномиальной задержкой) считается наиболее эффективным. Такой алгоритм на каждом шаге находит в точности одно решение и имеет временную оценку вида $O(N)$, где N - полином от размера входа задачи (полином от размеров матрицы m и n). Причем оценка дается для самой сложной индивидуальной задачи (для худшего случая). Требуемые алгоритмы удалось построить для немногих частных случаев дуализации, например, для случая, когда в исходной КНФ каждая элементарная дизъюнкция содержит не более двух переменных. В гиперграфовой постановке это случай графа, в матричной постановке случай, когда каждой строке булевой матрицы L не более двух единичных элементов [2]. Наилучший теоретический результат получен в 1995 г. Л. Хачияном с соавторами [3]. Построен алгоритм дуализации гиперграфа с квазиполиномиальной временной оценкой $O(N^{\log N})$, где N - полином от размера входа и выхода задачи (полином от m и n и числа решений, найденных на предыдущих шагах). Таким образом, статус дуализации в плане полиномиальной разрешимости до сих пор неизвестен, далее речь пойдет о сложности дуализации «в среднем» (для почти всех индивидуальных задач).

В 1977 г. Е. В. Дюковой предложен подход к построению асимптотически оптимальных алгоритмов дуализации [5]. Асимптотически оптимальные алгоритмы построены при условии, что число строк матрицы существенно меньше числа ее столбцов. Обоснование подхода опирается на технику получения асимптотических оценок числа неприводимых покрытий, которая первоначально была предложена в работах В.А. Слепян и В.Н. Носкова [11], а затем развита в работах Е.В. Дюковой [6] и А.Е. Андреева.

Целью работы является обзор алгоритмов, позволяющих решить поставленную задачу, а также их экспериментальное сравнение для булевых матриц разных размеров. Выбор оптимального переборного алгоритма для построения достаточно представительной выборки неприводимых покрытий позволит сократить время нахождения минимального покрытия. Ввиду невозможности построения всех неприводимых покрытий для матриц с большим числом столбцов в работе будут рассмотрены стохастические алгоритмы.

2 Постановка задачи

В матричной формулировке задача о покрытии может быть поставлена следующим образом [12].

Пусть дана булева матрица L размера $m \times n$, каждому столбцу которой приписано целое положительное число - вес. В качестве покрываемого множества $V = \{v_1, v_2, ..., v_m\}$ рассматривается совокупность строк матрицы L, покрывающее множество $E = \{e_1, e_2, ..., e_n\}$ состоит из столбцов матрицы L. Задан предикат $P(v_i, e_j)$ такой, что

$$P(v_i, e_j) = \begin{cases} 1 & \text{если на пересечении строки } v_i \text{ и столбца } e_j \text{ стоит } 1 \\ 0 & \text{иначе} \end{cases}$$

В случае, когда $P(v_i, e_j) = 1$, будем говорить, что столбец e_j покрывает строку v_i, а строка v_i покрывается столбцом e_j. Набор столбцов $H \subseteq E$ матрицы L называется покрытием, если каждая строка матрицы L покрывается хотя бы одним из столбцов, входящих в H. Требуется найти покрытие матрицы L с минимальной суммой весов.

В настоящей работе веса столбцов матрицы L полагаются равными 1. Требуется найти покрытие матрицы L минимальной мощности (длины). Такое покрытие называется минимальным. Покрытие называется неприводимым, если никакое его собственное подмножество не является покрытием. Поставленная задача поиска минимального покрытия булевой матрицы является труднорешаемой. В работе она решается приближенно с помощью стохастического перебора неприводимых покрытий.

Нетрудно видеть, что набор H из r столбцов матрицы L является неприводимым покрытием тогда и только тогда, когда выполнены следующие два условия:

1. Подматрица L^H матрицы L, образованная столбцами набора H, не содержит строки вида $(0, 0, ..., 0)$ (условие покрываемости);

2. L^H содержит каждую из строк вида $(1, 0, 0, ..., 0, 0), (0, 1, 0, ..., 0, 0), (0, 0, 0, ..., 0, 1)$, т.е. с точностью до перестановки строк содержит единичную подматрицу порядка r. Набор столбцов, удовлетворяющий данному условию, называется совместимым [4].

Приведенный критерий используется при построении множества неприводимых покрытий булевой матрицы.

3 Приближенные алгоритмы решения задачи о покрытии

3.1 Градиентный алгоритм

Известен эвристический метод решения задачи о покрытии, который носит название градиентного алгоритма (другие названия этого же метода: жадный алгоритм, метод наискорейшего спуска, метод вычерпывания). Метод заключается в пошаговом отборе в покрытие столбцов матрицы. На очередном шаге отбирается столбец, покрывающий наибольшее число строк, не покрытых к данному моменту. В следующей теореме приводится оценка длины минимального покрытия, построенного с помощью градиентного алгоритма [10].

Теорема 1 (Оценка длины покрытия, построенного градиентным алгоритмом). Пусть задано число $\alpha \in (0, 1]$ и булева матрица L размера $m \times n$. Пусть в любой строке содержится не менее αn единиц. Тогда длина покрытия, построенного с помощью градиентного алгоритма, не превосходит

$$\lfloor \left(\frac{1}{\alpha} \log_2^+(\alpha m) \right) \rfloor + \frac{1}{\alpha}, \log_2^+(x) = 1(x \geq 1) \log_2(x)$$

Заметим, что приведенная оценка быстро растет при $\alpha \to 0$. Действительно, экспериментально установлено, что градиентный алгоритм не эффективен в случае разреженных булевых матриц, которые довольно часто возникают в практических задачах. Однако он используется для нахождения верхней оценки длины минимального покрытия, позволяющей уменьшить перебор неприводимых покрытий для поиска минимального.

3.2 Стохастический алгоритм

В настоящей работе рассматривается алгоритм стохастического поиска минимального покрытия булевой матрицы, один шаг которого можно описать следующим образом. В зависимости от соотношения размеров матрицы L случайным образом выбирается ее подматрица M. Применяется градиентный алгоритм для нахождения верхней оценки длины минимального покрытия. Минимальное покрытие находится из множества неприводимых покрытий, построенных заданным детерминированным методом. Далее предложен псевдокод описанного шага стохастического алгоритма.

Algorithm 1 Шаг стохастического алгоритма

```
Require: Булева матрица L, детерминированный алгоритм перебора всех неприводимых покрытий A
  m ← L.shape[0]                          ▷ Число строк матрицы L
  n ← L.shape[1]                          ▷ Число столбцов матрицы L
  if m > n then
      M ← L                               ▷ Случайная подматрица из подмножества столбцов L
  else if m ≤ n then
      M ← L                               ▷ Случайная подматрица из подмножества строк L
  end if
  max_length ← Gradient(M)                ▷ Верхняя оценка длины
  P(M) ← A(M, max_length)                 ▷ Поиск неприводимых покрытий булевой матрицы
  min_cover ← min(P(M))                   ▷ Выбор минимального покрытия
  return min_cover
```

Описанный шаг осуществляется T раз. Число шагов T выбирается в зависимости от времени, выделяемого на работу стохастического алгоритма. Чем больше T, тем длина полученного приближенного минимального покрытия ближе к длине реального. После T шагов выполнение прекращается, отбирается то покрытие, которое имеет наименьшую длину.

Таким образом, в стохастических методах минимальное покрытие ищется по выборке из множества неприводимых покрытий булевой матрицы. Процесс перебора всех неприводимых покрытий случайной подматрицы ускоряется за счет использования верхней оценки длины минимального покрытия, полученной с помощью градиентного метода.

В научной литературе вопрос приближенного поиска множества всех неприводимых покрытий был поднят в 70-х годах прошлого века. Например, в теории распознавания идея стохастических тестовых алгоритмов принадлежит С.В. Яблонскому и В.Б. Кудрявцеву; она была реализована в работе В.Е.

Кузнецова [9]. Построенные в этой работе алгоритмы позволили увеличить площадь обрабатываемых матриц с обучающей информацией. Однако эти алгоритмы, так же как и первые детерминированные алгоритмы, были нацелены в основном на обработку матриц с относительно малым в сравнении с числом строк числом столбцов. Незначительное увеличение числа столбцов приводило к существенным вычислительным трудностям. На практике часто встречаются задачи, в которых число столбцов (признаков) значительно превосходит число строк (обучающих прецедентов). Для этого случая в [4] описаны эффективные алгоритмы, носящие стохастический характер.

Составной частью стохастического подхода являются детерминированные методы, поэтому построение стохастических алгоритмов стало возможным после решения задачи нахождения всех покрытий путем использования возможно меньшего перебора. Таким образом, скорость стохастического алгоритма во многом зависит от того, насколько эффективно работает его детерминированная часть. Подробнее остановимся на вопросе эффективности перечислительных алгоритмов.

4 Вычислительные эксперименты

В стохастическом подходе для построения выборки из множества неприводимых покрытий выделяется некоторое число случайных подматриц исходной матрицы L (См. 3.2). Так, в матрицах, в которых число строк превышает число столбцов, случайным образом выбираются подмножества столбцов; если же столбцов больше чем строк, то выбираются строки. Внутри полученных подматриц строится множество неприводимых покрытий.

В матрицах с преобладающим числом строк будем использовать перебор столбцов, упорядочив их следующим образом: каждому набору столбцов сопоставим двоичный набор, в котором единичным элементам соответствуют столбцы, вошедшие в набор, нулевым – все остальные. Перебираются все двоичные наборы с нулевого по единичный. Нетрудно заметить, что если некоторый набор столбцов образует покрытие, то рассмотрение охватывающих наборов не имеет смысла, поскольку они не могут образовывать неприводимых покрытий. Из построенной выборки берется покрытие наименьшей длины. В матрицах с преобладающим числом столбцов эффективным является применение алгоритма RUNC-M. Оба метода для повышения эффективности работы используют верхнюю оценку длины минимального покрытия, полученную с помощью градиентного метода. В классе квадратных матриц производится сравнение обоих алгоритмов с целью выявления наиболее эффективного.

4.1 Результаты эксперимента

На рисунках ниже представлено время работы алгоритмов перебора столбцов для матриц с преобладающим числом строк и RUNC-M для матриц с преобладающим числом столбцов. Для изучения динамики роста времени работы алгоритма перебора столбцов при увеличении их числа вычисления произведены для разных плотностей единичных элементов (плотности указаны в графе Density). При изменении числа строк плотность оставалась постоянной. При исследовании временные замеры усреднены по десяти запускам. Аналогичным образом производились временные замеры для алгоритма RUNC-M.

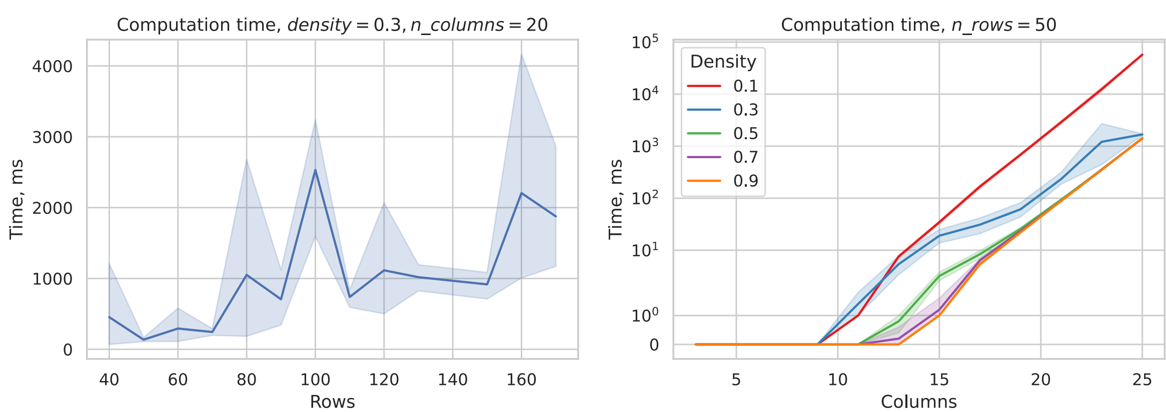

Рис. 1: Время работы алгоритма перебора в зависимости от числа строк при 20 столбцах (слева), время работы алгоритма перебора в зависимости от числа столбцов при 50 строках (справа)

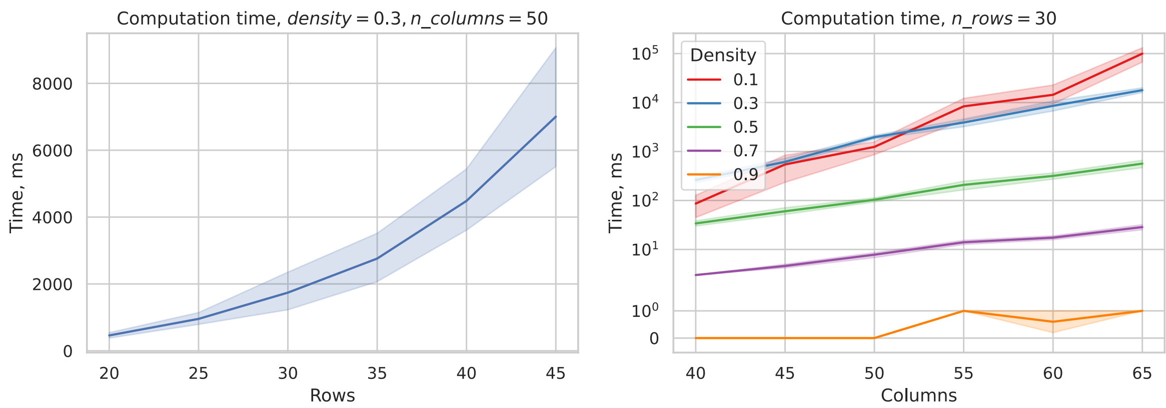

Рис. 2: Время работы RUNC-M в зависимости от числа строк при 50 столбцах (слева), время работы RUNC-M в зависимости от числа столбцов при 30 строках (справа)

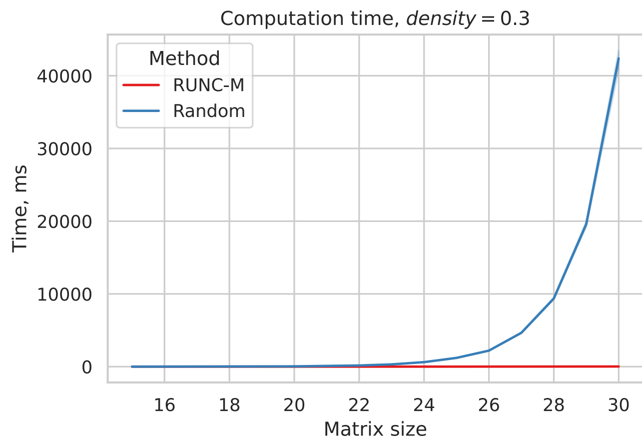

Рис. 3: Сравнение двух алгоритмов для класса квадратных матриц,
Random - алгоритм перебора столбцов; RUNC-M - поиск покрытий с помощью алгоритма RUNC-M

Это выводы эксперимента -

4.2 ~~Обсуждение~~ и выводы

Из Рис. 1 нетрудно заметить, что на подматрицах большего размера алгоритм работает дольше. Однако увеличение числа строк не так сильно сказывается на времени работы, как увеличение числа столбцов, при котором время работы растет экспоненциально. Также на время работы в значительной степени влияет плотность единичных элементов в матрице. Ожидаемо на низких плотностях перебор работает дольше, так как покрытий находится меньше, и, как следствие, больше наборов остается для перебора.

Как видно из Рис. 2, увеличение числа строк также приводит к росту времени работы алгоритма. Заметим, что при добавлении новых столбцов время работы RUNC-M растет медленнее, чем время работы алгоритма перебора. Как и в предыдущем эксперименте, оно сильно зависит от плотности единичных элементов.

На Рис. 3 приведено сравнение времен работы обоих алгоритмов в классе квадратных матриц. Заметим, что на матрицах, имеющих размер больше 20×20, время работы алгоритма перебора становится значительно больше времени работы RUNC-M. Таким образом, в классе квадратных матриц экспериментально установлена эффективность алгоритма RUNC-M.

Нужны выводы по всей статье (заключение).

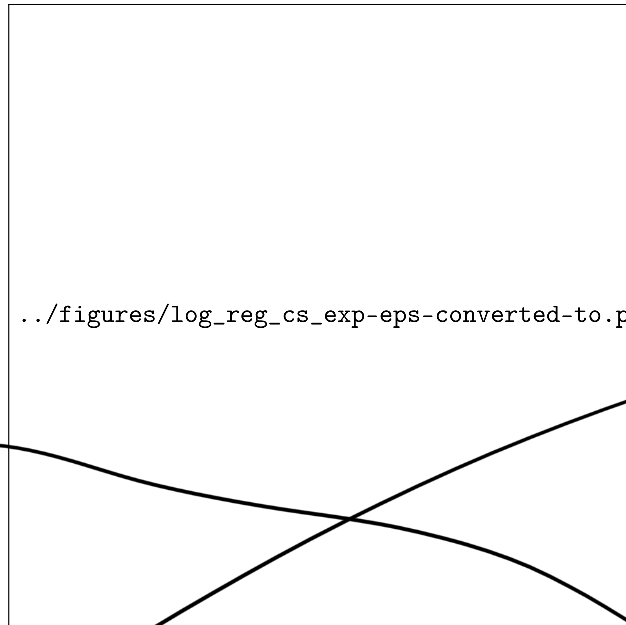

Рис. 4: Sample figure caption.

5 Examples of citations, figures, tables, references

5.1 Citations

Citations use `natbib`. The documentation may be found at

`http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf`

Here is an example usage of the two main commands (`citet` and `citep`): Some people thought a thing [? ?] but other people thought something else [?]. Many people have speculated that if we knew exactly why ?] thought this. . .

5.2 Figures

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. See Figure 4. Here is how you add footnotes. [1] Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

5.3 Tables

See awesome Table 1.

The documentation for `booktabs` (‘Publication quality tables in LaTeX’) is available from:

`https://www.ctan.org/pkg/booktabs`

[1] Sample of the first footnote.

Таблица 1: Sample table title

	Part	
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

5.4 Lists

- Lorem ipsum dolor sit amet
- consectetur adipiscing elit.
- Aliquam dignissim blandit est, in dictum tortor gravida eget. In ac rutrum magna.

Список литературы

[1] Alberto Caprara, Paolo Toth, and Matteo Fischetti. Algorithms for the set covering problem. Annals of Operations Research, 98(1-4):353–371, 2000.

[2] David S Johnson, Mihalis Yannakakis, and Christos H Papadimitriou. On generating all maximal independent sets. Information Processing Letters, 27(3):119–123, 1988.

[3] Leonid Khachiyan, Endre Boros, Khaled Elbassioni, and Vladimir Gurvich. An efficient implementation of a quasi-polynomial algorithm for generating hypergraph transversals and its application in joint generation. Discrete Applied Mathematics, 154(16):2350–2372, 2006.

[4] ЕВ Дюкова. Асимптотически оптимальные тестовые алгоритмы в задачах распознавания. Проблемы кибернетики, 39:165–199, 1982.

[5] Елена Всеволодовна Дюкова. Об асимптотически оптимальном алгоритме построения тупиковых тестов. In Доклады Академии наук, volume 233, pages 527–530. Российская академия наук, 1977.

[6] Елена Всеволодовна Дюкова and Петр Александрович Прокофьев. Об асимптотически оптимальных алгоритмах дуализации. Журнал вычислительной математики и математической физики, 55(5):895–910, 2015.

[7] Аркадий Дмитриевич Закревский. Логический синтез каскадных схем. 1981.

[8] П Кристофидес. Теория графов. Алгоритмический подход, 1978.

[9] ВЕ Кузнецов. Об одном стохастическом алгоритме вычисления информационных характеристик таблиц по методу тестов. Дискретный анализ, 23:8–23, 1973.

[10] Сергей Андреевич Ложкин. Лекции по основам кибернетики. М.: Издательский отдел ф-та ВМиК МГУ, 2004.

[11] ВН Носков and ВА Слепян. О числе тупиковых тестов для одного класса таблиц. Кибернетика. Киев, (1):60–65, 1972.

[12] ВЕ Тараканов. Комбинаторные задачи и (0, 1)-матрицы. 1985.