Новые асимптотически оптимальные алгоритмы дуализации

Дюкова Е. В. edjukova@mail.ru

Прокофьев П.А. p_prok@mail.ru

Москва, ВЦ РАН

Задача дуализации
(труднорешаемая перечислительная задача)

- Дана КНФ, реализующая монотонную булеву функцию $F(x_1, \dots, x_n)$ и состоящая из m элементарных дизъюнкций. Требуется построить сокращенную ДНФ D_F функции $F(x_1, \dots, x_n)$ (перечислить все максимальные конъюнкции функции F).
- Если в ДНФ D_F заменить знаки $\wedge$ и $\vee$ соответственно на знаки $\vee$ и $\wedge$, то получим КНФ для функции ***dual*** $F = \bar{F}(\bar{x}_1, \dots, \bar{x}_n)$.
- Другие формулировки дуализации используют понятия неприводимого покрытия булевой матрицы и минимального вершинного покрытия гиперграфа.

Тривиальный алгоритм

- **Этап 1.** Умножение логических скобок с использованием закона дистрибутивности.
- **Этап 2.** Упрощение ДНФ, полученной на этапе 1, за счет устранение повторяющихся конъюнкций и замены не элементарных конъюнкций на элементарные.
- **Этап 3.** Устранение элементарных поглощений с применением равенства X v XY= X.
- **Пример.**

$$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_2 \vee x_4)=$$

$$=(x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (x_1 \wedge x_3 \wedge x_2) \vee (x_1 \wedge x_3 \wedge x_4) \vee (x_2 \wedge x_1) \vee (x_2 \wedge x_1 \wedge x_4) \vee (x_2 \wedge x_3) \vee (x_2 \wedge x_3 \wedge x_4) =$$

$$=(x_1 \wedge x_2) \vee (x_1 \wedge x_4) \vee (x_2 \wedge x_3)$$

(при увеличении n на 1 время работы алгоритма возрастает в два раза).

Теоретические оценки сложности дуализации «в худшем случае»

- Эффективность алгоритма дуализации оценивается временем выполнения одного шага (на каждом шаге находится в точности одно решение). Временные оценки даются для самой сложной задачи (для худшего случая).
- Наилучший результат получен в 1995 г. Л. Хачияном с соавторами). Построен **инкрементальный алгоритм** с квазиполиномиальной временной оценкой $O(N^{\log N})$, где N – полином от размера входа и выхода задачи (*полином от* m , n и числа решений, найденных на предыдущих шагах).
- Для немногих частных случаев построены **алгоритмы с полиномиальными задержками,** имеющие временные оценки вида $O(N)$, где N – полином от размера входа задачи (*полином от* m , n).
- Вопрос о полиномиальной разрешимости дуализации является открытым.

Матричная формулировка дуализации

- Пусть $L = (a_{ij}), i = 1,2, \dots, m, j = 1,2, \dots, n,$ – булева матрица,
- Набор различных столбцов H матрицы L называется **неприводимым покрытием**, если выполнены два условия:
- **1)** подматрица, образованная столбцами из H, содержит каждую из строк (1, 0, 0, ..., 0 , 0), (0, 1, 0, ..., 0, 0), …, (0, 0, 0, …, 0, 1);
- **2)** каждая строка матрицы L в пересечении хотя бы с одним столбцом из H дает 1.
- Условие **1)** – **условие совместимости**. Если выполнено 1), то H – совместимый набор столбцов
- Условие **2)** – **условие покрываемости**. Если выполнено 2), то H – покрытие (H покрывает каждую строку матрицы L).
- Требуется построить (перечислить) все неприводимые покрытия матрицы L.

Теоретические оценки сложности дуализации «в среднем»

- В 1977 г. (Е.В. Дюкова) предложен подход к построению **асимптотически оптимальных алгоритмов** дуализации (алгоритмов, эффективных в среднем).
- Асимптотически оптимальные алгоритмы построены для случая $log m \le (1-\varepsilon) log n$, $0 < \varepsilon < 1$, и отличаются от алгоритмов с полиномиальными задержками тем, что имеют «лишние» полиномиальные шаги.
- На каждом шаге строится максимальный совместимый набор столбцов H матрицы L (совместимый набор столбцов H – максимальный, если не существует другого совместимого набора столбцов, содержащего H).
- Неприводимое покрытие матрицы L – максимальный совместимый набор столбцов. Обратное неверно, так как может быть не выполнено условие покрываемости.

Основные виды асимптотически оптимальных алгоритмов

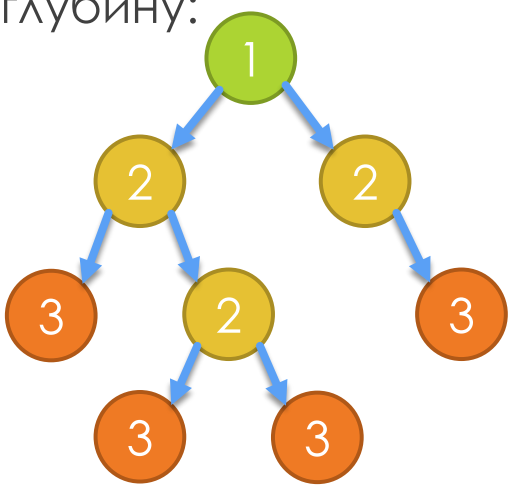
- **Шаг считается лишним в двух случаях:** 1)построенный набор столбцов не является покрытием; 2)построенный набор столбцов уже строился на предыдущих шагах. Лишний шаг определяется за полиномиальное время от размера входа.
- Существуют **два вида асимптотически оптимальных алгоритмов:** алгоритмы с повторяющимися шагами и алгоритмы без повторяющихся шагов.
- **Основное требование:** для почти всех булевых матриц размера $m \times n$ число лишних шагов должно иметь более низкий порядок роста по сравнению с числом неприводимых покрытий при росте размера задачи (при $m, n \to \infty$) .

Экспериментально эффективные алгоритмы дуализации

- *В работах Murakami K., Uno T. (2011- 2013 гг.) построены алгоритмы, которые являлись до настоящего времени* лидерами по скорости счета. При описании алгоритмов использовались понятия теории гиперграфов
- Принципиальная схема работы японских алгоритмов не отличается от схемы работы асимптотически оптимальных алгоритмов, построенных ранее в наших работах (в основе японских алгоритмов лежит условие «crit», которое эквивалентно условию совместимости набора столбцов булевой матрицы) .
- Наши алгоритмы работали медленнее. Была поставлена задача усовершенствования их конструкций с целью получения лучших, чем у японцев результатов.

Дерево решений асимптотически оптимального алгоритма

- Работу асимптотически оптимального алгоритма дуализации можно представить в виде обхода дерева решений (ДР) в глубину:



1 — пустой набор;

2 — совместимый набор столбцов;

3 — максимальный совместимый набор столбцов, который либо является неприводимым покрытием, либо соответствует лишнему шагу.

- Каждая дочерняя вершина внутренней вершины H образуется добавлением к H в точности одного столбца.
- Время работы алгоритма зависит от числа вершин в ДР, которое строит этот алгоритм.

Алгоритмы АО2 и ОПТ

- Первоначально наши усилия были направлены в основном на сокращение числа лишних шагов за счет усложнения шага. Среди алгоритмов без повторений **л**идером был алгоритм ОПТ (Е.В. Дюкова, А.С. Инякин, 2008 г.). Этот алгоритм **имеет сложный шаг с** оценкой $O\big(qmn(m+q)\big)$, где $q = \min\{m, n\}$.
- Среди алгоритмов с повторяющимися шагами лидером являлся алгоритм АО2 (Е.В. Дюкова, , 2004 г.), с той же сложностью, что и ОПТ. Алгоритм АО2 на каждом шаге за полиномиальное время строит неприводимое покрытие. Однако повторяет решения, найденные ранее.
- Усовершенствование алгоритма АО2 (на основе уменьшения числа промежуточных вычислений на шаге без изменения сложности шага) привело к уменьшению сложности ДР и максимальному сокращению времени счета примерно в 4 раза (Е.В. Дюкова, П.А. Прокофьев, 2014 г.).

Новые алгоритмы дуализации RUNC, RUNC-M и PUNC

- Более существенные результаты были получены при усовершенствовании алгоритма ОПТ [***Дюкова Е.В., Прокофьев П.А.,*** **Построение и исследование новых асимптотически оптимальных алгоритмов дуализации // Машинное обучение и анализ данных, 2014].**
- Были применены конструктивные приемы, которые ранее использовались для модификации алгоритма **АО2.**
- В результате были построены новые алгоритмы дуализации **RUNC**, **RUNC-M** и **PUNC**, которые выгодно отличаются от **ОПТ** следующими показателями: 1) число вершин в ДР значительно меньше; 2) сложность шага равна $O(qmn)$ (меньше сложности шага **ОПТ**).

Наилучший алгоритм дуализации RUNC-M

- Наилучшие результаты показал алгоритм **RUNC-M**.
- В алгоритме **RUNC-M** применяется «жадная» стратегия построения ДР:
  - всякий раз после построения очередной вершины H выбирается строка i, непокрытая набором столбцов H и содержащая наименьшее число единиц в пересечении со столбцами, которые совместимы с H (столбец h совместим с набором столбцов H, если $\{H \cup h\}$ – совместимый набор столбцов);
  - каждая дочерняя для H вершина строится путём добавления к H столбца, покрывающего строку i и совместимого с H .

Тестовые задачи

- Тестирование новых и ранее построенных алгоритмов дуализации проводилось на задачах трех принципиально разных типов:
  1) случайные матрицы различного размера с равновероятным появлением 0 и 1;
  2) модельные данные различного типа и размера;
  3) прикладные задачи большого размера.
- Всего обсчитывалось около 1000 матриц.
- Модельные и прикладные данных полностью совпадали с тестовыми данными, на которых осуществляли тестирование своих алгоритмов японские учёные.

Результаты экспериментов

- Практически на всех задачах лидирует один из новых алгоритмом. Исключением являются два типа модельных задач (порядка 20 матриц), ключевой характеристикой которых является сильная разреженность матриц.
- Наилучшие результаты показывает алгоритм RUNC-M. Причем на некоторых задачах большого размера скорость RUNC-M превосходит скорость ближайшего соперника по меньшей мере в два раза, а иногда и в 5 раз.
- Счет, проведенный в настоящей работе и в работе японских авторов, демонстрирует преимущество асимптотически оптимальных алгоритмов дуализации перед с алгоритмами, имеющими другие приципы конструирования. Асимптотически оптимальным алгоритмам значительно уступают инкрементальные алгоритмы.

Число вершин в ДР

Модельная задача dualmatching					
ОПТ	39375	354305	3188659	9565952	28697829
RUNC	**1033**	**4107**	**16397**	**32782**	**65551**
RUNC-M	**1033**	**4107**	**16397**	**32782**	**65551**
PUNC	1034	4108	16398	32783	65552
m	1024	4096	16384	32768	65536
n	21	25	29	31	33
|P(L)|	10	12	14	15	16

Прикладная задача lose					
ОПТ	18381	218422	568938	1614743	18803203
RUNC	6937	59228	120265	326222	990441
RUNC-M	**5039**	**44205**	**69178**	**174305**	**462764**
PUNC	6570	52171	105525	248472	664007
m	100	200	400	800	1600
n	77	77	79	81	81
|P(L)|	2341	22760	33087	79632	212761

Время счёта (лидирует RUNC-M)

Прикладная задача accidents						
AO1	25.3	-	-	-	-	-
AO2	0.11	5.4	68.7	-	-	-
AO2M	0.05	1.7	23.1	175.6	-	-
ОПТ	**0.001**	0.05	0.23	1.1	8.2	136.3
MMCS	0.016	0.06	0.27	0.8	3.7	33.
RS	**0.001**	0.05	0.3	0.8	3.4	35.3
RUNC	0.031	0.031	0.09	**0.25**	**0.9**	7.6
RUNC-M	**0.001**	**0.016**	**0.08**	**0.25**	**0.9**	**7.3**
PUNC	**0.001**	0.031	0.27	1.4	9.9	208.3
m	447	2000	4322	10968	32207	135439
n	64	81	336	336	336	442
|P(L)|	1039	3547	7617	17486	47137	185218

Время счёта (лидирует RUNC-M)

Прикладная задача lose							
AO1	-	-	-	-	-	-	-
AO2	-	-	-	-	-	-	-
AO2M	2.7	28.7	325.9	-	-	-	-
ОПТ	1.3	4.2	66.5	-	-	-	-
MMCS	0.11	0.3	1.1	12.9	41.8	185.8	519.5
RS	0.16	0.5	1.7	17.1	107.1	449.7	-
RUNC	0.09	0.5	2.2	10.3	58.5	278.6	-
RUNC-M	**0.05**	**0.17**	**0.5**	**4.3**	**12.5**	**49.2**	**137.7**
PUNC	0.4	1.5	5.8	71.1	501.2	-	-
m	400	800	1600	3200	6400	12800	16635
n	79	81	81	81	81	85	85
|P(L)|	33087	79632	212761	2396735	4707877	16405082	39180611

Время счёта (лидирует алгоритм японских учёных)

Модельная задача self dual threshold hypergraph								
AO2M	-	-	-	-	-	-	-	-
ОПТ	103.2	217.8	411.1	-	-	-	-	-
MMCS	**0.05**	0.08	**0.09**	**0.14**	**0.23**	0.4	**0.5**	0.7
RS	**0.05**	**0.06**	**0.09**	0.16	**0.23**	**0.3**	0.6	**0.6**
RUNC	0.08	0.11	0.19	0.4	0.7	1.4	2.	3.1
RUNC-M	0.09	0.12	0.23	0.4	0.9	1.3	2.	3.2
PUNC	0.5	0.7	1.7	2.2	5.7	20.9	12.7	27.6
m	6482	8192	10102	14522	19742	25762	32582	40202
n	163	183	203	243	283	323	363	403
|P(L)|	6482	8192	10102	14522	19742	25762	32582	40202

Средняя скорость работы алгоритмов дуализации