Новый алгоритм для задачи выполнения наибольшего количества дизъюнктов

Алфёров В. В.

Научный руководитель: к.ф.-м.н. Близнец И. А.

Санкт-Петербургская школа физико-математических и компьютерных наук

НИУ ВШЭ – Санкт-Петербург

Санкт-Петербург, 2020

Задача булевой выполнимости

- Литерал булевой переменной x — выражение x или $\overline{x}$.
- Дизъюнкт — дизъюнкция литералов, то есть выражение вида $x \vee \overline{y} \vee z$.
- Формула в конъюнктивной нормальной форме (КНФ) — конъюнкция дизъюнктов.
- Задача булевой выполнимости (SAT): можно ли найти хотя бы какие-то значения переменных, при которых формула выполняется (т.е. принимает значение 1)?

Пример невыполнимой формулы:

$$\overline{x} \quad \wedge \quad \overline{y} \quad \wedge \quad \overline{z} \quad \wedge \quad (x \vee y) \quad \wedge \quad (y \vee z) \quad \wedge \quad (x \vee z)$$

Задача максимальной выполнимости

Задача максимальной выполнимости (MAXSAT): какое наибольшее количество дизъюнктов можно выполнить?

Пример:

$$\boxed{\overline{x}} \wedge \overline{y} \wedge \overline{z} \wedge \boxed{(x \vee y) \wedge (y \vee z) \wedge (x \vee z)}$$

Четыре дизъюнкта выполняются при $x = 0, y = z = 1$.

Это одна из первых NP-полных задач.

На практике NP-полные задачи часто решают сведением к MAXSAT через солверы (например, RC2[1]).

[1] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. "RC2: An efficient MaxSAT solver". In: (2019).

Обозначения и вариации

Обозначения:

- Длина дизъюнкта – это количество литералов в нём.
- Длина формулы L – это количество литералов в ней.
- k-переменная – переменная, входящая k раз.
- n_k – количество k-переменных в формуле.

Вариации задачи:

- MAX-k-SAT — длина каждого дизъюнкта не более k.
- (n, k)-MAXSAT — переменные входят не более k раз.

Даже задача $(n, 3)$-MAX-2-SAT уже NP-полна[2].
Задача $(n, 2)$-MAXSAT решается за полиномиальное время.

[2] Venkatesh Raman, Bala Ravikumar, and S Srinivasa Rao. "A simplified NP-complete MAXSAT problem". In: (1998).

Расщепление по переменной

Расщепление по x – это расщеплление на случаи $x = 0$ и $x = 1$.

Пример:

Расщепимся на случаи $x = 0$ и $x = 1$ в формуле

$$(x \vee y) \wedge (x \vee \overline{z}) \wedge (\overline{x} \vee w \vee \overline{y}) \wedge (\overline{x} \vee z) \wedge \dots$$

$$\xrightarrow{x = 0} \quad y \wedge \overline{z} \wedge \dots$$

$$\xrightarrow{x = 1} \quad (w \vee \overline{y}) \wedge z \wedge \dots$$

В первом случае длина уменьшилась на 7, во втором – на 6.

Предыдущие результаты

Автор	Год	k	m	L
Niedermeier et al.	1999	$O^*(1.40^k)$	$O^*(1.38^m)$	$O^*(1.1279^L)$
Bansal et al.	1999	$O^*(1.38^k)$	$O^*(1.34^m)$	$O^*(1.1058^L)$
Chen et al.	2004	$O^*(1.37^k)$	$O^*(1.33^m)$	–
Bliznets et al.	2012	$O^*(1.36^k)$	–	–
Chen et al.	2015	$O^*(1.33^k)$	–	–
Xu et al.	2019	–	$O^*(1.30^m)$	–

Легенда:

- k – Оценка относительно ответа.
- m – Оценка относительно количества дизъюнктов.
- L – Оценка относительно длины формулы.

Меры сложности задачи

- Теоретические оценки отстают от практических результатов.
- Развитие алгоритмов от разных мер позволяет смотреть на задачу под разными углами:
  - Если есть много длинных дизъюнктов, задачу выгоднее решать относительно количества дизъюнктов или ответа.
  - Если есть много дизъюнктов длины 1 или 2, задачу выгоднее решать относительно длины.
- Последние результаты улучшают оценки относительно количества выполненых или всех дизъюнктов.

В данной работе строится алгоритм, улучшающий оценку относительно длины.

Недостатки меры длины

Дисбалланс весов переменных:

- Переменные, встречающиеся менее, чем дважды, можно сразу убрать.
- Значит, если у 3-переменной убрать хотя бы один литерал, она сразу исчезает, длина уменьшается на 3.
- Если у 4-переменной убрать один литерал, длина уменьшается на 1.

Благодаря этому $(n, 3)$-MAXSAT решается за $O^*(1.0600^L)$.[3]

Выходит, что пока в формуле есть 4-переменные, алгоритм работает за $O^*(1.106^L)$, а затем остаток решается быстрее.

Другая похожая мера может помочь провести более аккуратный анализ.

[3] Tatiana Belova and Ivan Bliznets. "Upper and Lower Bounds for Different Parameterizations of (n, 3)-MAXSAT". In: 2018.

Цель и задачи

Цель: Улучшить алгоритм относительно длины для задачи MAXSAT.

Задачи:

- Разработать меру сложности задачи, сглаживающую разницу между 3-переменными и 4-переменными.
- Разобрать переменные с большим количеством вхождений.
- Применить разработанную меру для разбора 4- и 5-переменных.

Уменьшенная мера

Была выбрана мера $d = L - n_3$.

Свойства:

- Вес 3-переменных уменьшен.
- Так как $d \leq L$, алгоритм за $O^*(\alpha^d)$ работает и за $O^*(\alpha^L)$.
- Для $(n, 3)$-MAXSAT: $L = 3n_3 = \frac{3}{2}d$, и алгоритм работает за $O^*(1.0600^{\frac{3}{2}d}) = O^*(1.0912^d)$
- На переменных с большим количеством вхождений ведёт себя как длина.

А именно, при расщеплении по такой переменной мера уменьшается не меньше, чем длина.

Переменные с большим количеством вхождений

У переменных с шестью и более вхождениями много соседей.

При расщеплении по ним длина (следовательно, и d) уменьшается сильно.

Конкретнее, оказывается, что расщепление по такой переменной даёт алгоритм за $O^*(1.0927^d)$.

Вывод: Случай переменных с большим количеством вхождений разобран.

Разбор 5-переменных

Пример формулы:

$$(x \vee C_1) \wedge (x \vee C_2) \wedge (x \vee C_3) \wedge (\overline{x} \vee D_1) \wedge (\overline{x} \vee D_2) \wedge F'$$

Мера: $d = L - n_3 = 2n_3 + 4n_4 + 5n_5$

- Почти всегда d уменьшается сильнее, чем длина.
- Если уменьшается также, то вокруг много 5-переменных.
- Тогда можно вывести более сложные расщепления.
- В алгоритме за $O^*(1.0927^d)$ разбор 16 случаев.

Разбор 4-переменных

Пример формулы:

$$(x \lor C_1) \land (x \lor C_2) \land (\overline{x} \lor D_1) \land (\overline{x} \lor D_2) \land F'$$

Мера: $d = L - n_3 = 2n_3 + 4n_4$

- При удалении одного литерала любой переменной d уменьшается на 2.
- Таким образом выравниваются 3-переменные и 4-переменные.
- Это позволяет разбирать 4-переменные почти также быстро, как 3-переменные.
- В алгоритме за $O^*(1.0927^d)$ разбор 12 случаев.

Вывод: Случай переменных с четырьмя или пятью вхождениями разобран.

Результаты

Разработан алгоритм за $O^*(1.0927^L)$ для задачи MAXSAT (улучшение на 11.5%).

- Предложена мера $d = L - n_3$, сглаживающая разницу между 3-переменными и 4-переменными.
- Показано, что на переменных с большим количеством вхождений мера ведёт себя как длина.
- С помощью свойств меры на 4- и 5-переменных разобраны переменные с небольшим количеством вхождений.

Работу планируется подать на конференцию AAAI-21.

MAXSAT относительно ответа

Автор	Год	Время работы	Δ
Niedermeier et al.	1999	$O^*(1.3995^k)$	–
Bansal et al.	1999	$O^*(1.3803^k)$	4.1%
Chen et al.	2004	$O^*(1.3695^k)$	2.5%
Bliznets et al.	2012	$O^*(1.3579^k)$	2.7%
Chen et al.	2015	$O^*(1.3248^k)$	8.1%

MAXSAT относительно количества дизъюнктов

Работа	Год	Результат	Δ
Niedermeier et al.	1999	$O^*(1.3803^m)$	–
Bansal et al.	1999	$O^*(1.3413^m)$	8.9%
Chen et al.	2004	$O^*(1.3248^m)$	4.2%
Xu et al.	2019	$O^*(1.2989^m)$	7%

Алгоритмы для $(n, 3)$-MAXSAT

Работа	Год	Результат
Raman et al.	1998	$O^*(1.2010^L)$
Bansal et al.	1999	$O^*(1.0983^L)$
Kulikov et al.	2009	$O^*(1.0836^L)$
Bliznets	2013	$O^*(1.0801^L)$
Chen et al.	2015	$O^*(1.0735^L)$
Li et al.	2017	$O^*(1.0609^L)$
Belova et al.	2018	$O^*(1.0600^L)$

Расщепление по переменной: доказательство

- Можно записать изменение уменьшенной меры как сумму уменьшений весов всех переменных.
- У переменных с ≥ 4 вхождениями вес изменился не меньше, чем длина.
- У 3-переменной, у которой пропали не все литералы, вес изменился на 2.
- 3-переменную, у которой пропадает 3 литерала, можно предварительно сократить.[4]

[4] Chao Xu et al. "Resolution and domination: an improved exact MaxSAT algorithm". In: 2019.