(n, 3)-MAX-SAT

Татьяна Белова
Научный руководитель: Иван Близнец

Санкт-Петербургский Академический Университет

20.02.2018

Описание задачи

- Дана формула в КНФ. Каждая переменная входит не более трех раз
$(x \vee y \vee z) \wedge (\bar{x} \vee z) \wedge \bar{y} \wedge \bar{z}$
- Требуется определить, какое максимальное число клозов можно выполнить

Описание задачи

- Дана формула в КНФ. Каждая переменная входит не более трех раз
  $(x \vee y \vee z) \wedge (\bar{x} \vee z) \wedge \bar{y} \wedge \bar{z}$
- Требуется определить, какое максимальное число клозов можно выполнить
- Параметризации:
  - Относительно числа переменных – n
  - Относительно числа выполненных клозов – k

Мотивация

- $(n, 3)$-MAX-SAT – NP-трудная задача

Мотивация

- $(n, 3)$-*MAX-SAT* – *NP*-трудная задача
- $(n, 3)$-*MAX-SAT* – частный случай задачи *MAX-SAT*

Мотивация

- $(n, 3)$-*MAX-SAT* – *NP*-трудная задача
- $(n, 3)$-*MAX-SAT* – частный случай задачи *MAX-SAT*
- *MAX-SAT* применяется:
  - Искусственный интеллект
  - Комбинаторная оптимизация
    - Максимальный разрез
    - Минимальное вершинное покрытие
    - Максимальное независимое множество

Известные результаты

Time	Reference	Year
$O^*(1.732^n)$	Raman, Ravikumar, Rao	1998
$O^*(1.3248^n)$	Bansal, Raman	1999
$O^*(1.27203^n)$	Kulikov	2005
$O^*(1.2600^n)$	Bliznets	2013
$O^*(1.237^n)$	Xu, Chen, Wang	2016
$O^*(1.2207^n)$	Belova, Bliznets, Kopeliovich	2017

Time	Reference	Year
$O^*(1.3247^k)$	Chen, Kanj	2002
$O^*(1.2721^k)$	Bliznets, Golovnev	2012
$O^*(1.194^k)$	Xu, Chen, Wang	2016

*Известные результаты на начало семестра

Идея решения

- Если k сильно больше, чем n, то можно использовать алгоритм относительно n
- Иначе воспользуемся тем, что k маленькое

$(\nu(F) + k)$-SAT:

$(\nu(F) + k)$-SAT:

- По формуле F строим двудольный граф B_F. Первая доля – переменные, вторая – клозы.

$(\nu(F) + k)$-SAT:

- По формуле F строим двудольный граф B_F. Первая доля – переменные, вторая – клозы.
- Обозначим за $\nu(F)$ размер максимального паросочетания в B_F.

$(\nu(F) + k)$-SAT:

- По формуле F строим двудольный граф B_F. Первая доля – переменные, вторая – клозы.
- Обозначим за $\nu(F)$ размер максимального паросочетания в B_F.
- Хотим выполнить $(\nu(F) + k)$ клозов.

$(\nu(F) + k)$-*SAT*

$(\nu(F) + k)$-*SAT*:

- По формуле F строим двудольный граф B_F. Первая доля – переменные, вторая – клозы.
- Обозначим за $\nu(F)$ размер максимального паросочетания в B_F.
- Хотим выполнить $(\nu(F) + k)$ клозов.

Crowston et al.'12 свели $(\nu(F) + k)$-*SAT*
к задаче $(m - k)$-*Hitting-Set* и решили ее за
$O((2e)^{2k+O(\log^2 k)}(m+n)^{O(1)})$

$(\nu(F)+k)$-SAT

$(\nu(F)+k)$-SAT:

- По формуле F строим двудольный граф B_F. Первая доля – переменные, вторая – клозы.
- Обозначим за $\nu(F)$ размер максимального паросочетания в B_F.
- Хотим выполнить $(\nu(F)+k)$ клозов.

Crowston et al.'12 свели $(\nu(F)+k)$-SAT
к задаче $(m-k)$-*Hitting-Set* и решили ее за
$O((2e)^{2k+O(\log^2 k)}(m+n)^{O(1)})$

Francis et al.'13 решили $(m-k)$-*Hitting-Set* за $O^*(4^k)$.

Ход решения

- Упростим формулу и добьемся $\nu(F) = n$.

Ход решения

- Упростим формулу и добьемся $\nu(F) = n$.
- Подберем константу c и разберем два случая:
  - $k < c \cdot n \Rightarrow$ запускаем алгоритм для $(\nu(F) + k)$-SAT
  - $k > c \cdot n \Rightarrow$ запускаем алгоритм для $(n, 3)$-MAX-SAT

Ход решения

- Упростим формулу и добьемся $\nu(F) = n$.
- Подберем константу c и разберем два случая:
  - $k < c \cdot n \Rightarrow$ запускаем алгоритм для $(\nu(F) + k)$-SAT
  - $k > c \cdot n \Rightarrow$ запускаем алгоритм для $(n, 3)$-MAX-SAT

Пусть $(\nu(F) + k)$-SAT допускает алгоритм за c_1^k
Пусть $(n, 3)$-MAX-SAT допускает алгоритм за c_2^n

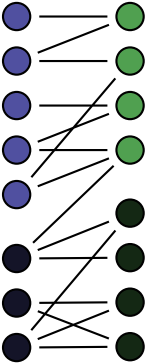
Оптимальная константа достигается при равенстве времен $c_1^{c \cdot n - n} = c_2^n \Leftrightarrow c = \frac{\log(c_1) + \log(c_2)}{\log(c_1)}$

Упрощение формулы

Как добиться $\nu(F) = n$?

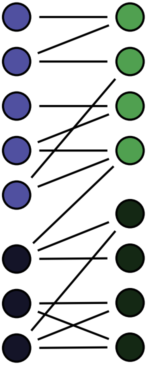
Как добиться $\nu(F) = n$?

- Найдем минимальное по включению подмножество X вершин левой доли, для которого не выполняется условие Холла (полиномиальное время)



Как добиться $\nu(F) = n$?

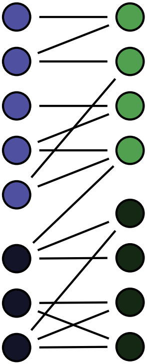
- Найдем минимальное по включению подмножество X вершин левой доли, для которого не выполняется условие Холла (полиномиальное время)
- Выполним соответствующие $N(X)$ клозы



Упрощение формулы

Как добиться $\nu(F) = n$?

- Найдем минимальное по включению подмножество X вершин левой доли, для которого не выполняется условие Холла (полиномиальное время)
- Выполним соответствующие $N(X)$ клозы
- Тем самым свелись к меньшей задаче с $n' = n - |X|$, $k' = k - |N(X)|$



Оценка времени работы

$(\nu(F)+k)$-SAT допускает алгоритм за c_1^k
$(n,3)$-MAX-SAT допускает алгоритм за c_2^n

$$c = \frac{\log(c_1)+\log(c_2)}{\log(c_1)}$$

Оценка времени работы

$(\nu(F) + k)$-SAT допускает алгоритм за c_1^k
$(n, 3)$-MAX-SAT допускает алгоритм за c_2^n

$c = \frac{\log(c_1)+\log(c_2)}{\log(c_1)}$

Получим время работы $T(k) \leq (\sqrt[c]{c_2})^k$

Оценка времени работы

$(\nu(F)+k)$-SAT допускает алгоритм за c_1^k
$(n,3)$-MAX-SAT допускает алгоритм за c_2^n

$c = \frac{\log(c_1)+\log(c_2)}{\log(c_1)}$

Получим время работы $T(k) \leq (\sqrt[c]{c_2})^k$

Подставим $c_1 = 4$, $c_2 = 1.220744$, получим:

- $c = 1.14388$
- $T(k) \leq 1.19050^k$

Это лучше предыдущего результата 1.194^k.

Результаты

Time	Reference	Year
$O^*(1.732^n)$	Raman, Ravikumar, Rao	1998
$O^*(1.3248^n)$	Bansal, Raman	1999
$O^*(1.27203^n)$	Kulikov	2005
$O^*(1.2600^n)$	Bliznets	2013
$O^*(1.237^n)$	Xu, Chen, Wang	2016
$\mathbf{O^*(1.2207^n)}$	**Belova, Bliznets, Kopeliovich** *	**2017**

$O^*(1.3247^k)$	Chen, Kanj	2002
$O^*(1.2721^k)$	Bliznets, Golovnev	2012
$O^*(1.194^k)$	Xu, Chen, Wang	2016
$\mathbf{O^*(1.19050^k)}$	**Belova, Bliznets** *	**2017**

Результаты

Time	Reference	Year
$O^*(1.732^n)$	Raman, Ravikumar, Rao	1998
$O^*(1.3248^n)$	Bansal, Raman	1999
$O^*(1.27203^n)$	Kulikov	2005
$O^*(1.2600^n)$	Bliznets	2013
$O^*(1.237^n)$	Xu, Chen, Wang	2016
$O^*(1.2207^n)$	Belova, Bliznets, Kopeliovich *	2017
$\mathbf{O^*(1.194^n)}$	**Li, Xu, Wang, Yang**	**2017**

$O^*(1.3247^k)$	Chen, Kanj	2002
$O^*(1.2721^k)$	Bliznets, Golovnev	2012
$O^*(1.194^k)$	Xu, Chen, Wang	2016
$O^*(1.19050^k)$	Belova, Bliznets *	2017
$\mathbf{O^*(1.175^k)}$	**Li, Xu, Wang, Yang**	**2017**

Результаты

Time	Reference	Year
$O^*(1.732^n)$	Raman, Ravikumar, Rao	1998
$O^*(1.3248^n)$	Bansal, Raman	1999
$O^*(1.27203^n)$	Kulikov	2005
$O^*(1.2600^n)$	Bliznets	2013
$O^*(1.237^n)$	Xu, Chen, Wang	2016
$O^*(1.2207^n)$	Belova, Bliznets, Kopeliovich *	2017
$O^*(1.194^n)$	Li, Xu, Wang, Yang	2017

$O^*(1.3247^k)$	Chen, Kanj	2002
$O^*(1.2721^k)$	Bliznets, Golovnev	2012
$O^*(1.194^k)$	Xu, Chen, Wang	2016
$O^*(1.19050^k)$	Belova, Bliznets *	2017
$O^*(1.175^k)$	Li, Xu, Wang, Yang	2017
$\mathbf{O^*(1.1701^k)}$	**Belova, Bliznets** *	**2017**

Результаты

Time	Reference	Year
$O^*(1.732^n)$	Raman, Ravikumar, Rao	1998
$O^*(1.3248^n)$	Bansal, Raman	1999
$O^*(1.27203^n)$	Kulikov	2005
$O^*(1.2600^n)$	Bliznets	2013
$O^*(1.237^n)$	Xu, Chen, Wang	2016
$O^*(1.2207^n)$	Belova, Bliznets, Kopeliovich *	2017
$O^*(1.194^n)$	Li, Xu, Wang, Yang	2017
$\mathbf{O^*(1.1907^n)}$	**Belova, Bliznets** *	**2017**

$O^*(1.3247^k)$	Chen, Kanj	2002
$O^*(1.2721^k)$	Bliznets, Golovnev	2012
$O^*(1.194^k)$	Xu, Chen, Wang	2016
$O^*(1.19050^k)$	Belova, Bliznets *	2017
$O^*(1.175^k)$	Li, Xu, Wang, Yang	2017
$\mathbf{O^*(1.1701^k)}$	**Belova, Bliznets** *	**2017**
$\mathbf{O^*(1.1677^k)}$	**Belova, Bliznets** *	**2017**