УДК 519.854

ВЫЧИСЛИТЕЛЬНЫЙ ЭКСПЕРИМЕНТ ПО ОЦЕНКЕ АЛГОРИТМИЧЕСКОЙ СЛОЖНОСТИ АНАЛИЗА ВЫПОЛНИМОСТИ/НЕВЫПОЛНИМОСТИ 3-КНФ БУЛЕВЫХ ФОРМУЛ

С.И. Уваров

Институт проблем управления им. В.А. Трапезникова РАН

Россия, 117997, Москва, Профсоюзная ул., 65

E-mail: uvarov53@gmail.com

Ключевые слова: алгоритмическая сложность, выполнимость, SAT, литерал, дизъюнкт, конъюнктивная нормальна форма, 3-КНФ.

Аннотация: Проведен вычислительный эксперимент по оценке алгоритмической сложности задачи выполнимости булевых формул, заданных в 3-КНФ. В эксперименте использован известный генератор случайных 3-КНФ формул и полный алгоритм анализа выполнимости таких формул, построенный на симбиозе DP и DPLL процедур. Потенциально неполиномиальная компонента оценки алгоритмическщй сложности определяется максимальным количеством ветвлений при выводе одного из дизъюнктов каждой из исследуемых 3-КНФ формул.

1. Введение

Задача выполнимости 3-КНФ булевых формул является опорной для оценки алгоритмической сложности NP-complete задач. Если ее сложность окажется экспоненциальной, субэкспоненциальной или полиномиальной, такой же сложностью будут характеризоваться и все остальные задачи класса NP. Оценка алгоритмической сложности задачи выполнимости 3-КНФ представима в виде

$$L(N) = E(N)P_1(N) + P_2(N).$$

$L(N)$– трудоемкость решения задачи, имеющей N переменных (для SAT-3 число переменных полиномиально связано с максимально возможной длиной 3-КНФ формулы) выраженна в некоторых элементарных действиях. $P_1(N)$ и $P_2(N)$ – некие полиномы от N. $E(N)$ – сомножитель, представляющий максимальный интерес. Часто его представляют в виде $E(N) = \alpha(N)^N$. Если в пределе (при $N \to \infty$) основание $\alpha(N)$ экспоненты больше единицы, оценку сложности следует признать экспоненциальной. Если в пределе и только в пределе $\alpha(N) = 1$, оценка сложности субэкспоненциальна. Примером субэкспоненциальной функции является $2^{\sqrt{N}}$. Если окажется, что существует число n^* такое, что при $N > n^*$ всегда $\alpha(N) = 1$, искомую оценку следует считать полиномиальной.

2. Генерация 3-КНФ формул с использованием датчика случайных чисел

Обозначим через $\mathbf{X}$ множество из N булевых переменных $X_\imath$, $\mathbf{X}$=$\{X_1, ..., X_N\}$. Литералами переменной $X_\imath$ назовем термы x_i^0 и x_i^1 ($x_i^0 = X_i$, $x_i^1 = \neg X_i$).

Исследуются формулы F_M, представленные в конъюнктивной нормальной форме, являющиеся конъюнкцией набора из M трехлитеральных дизъюнктов C_j^3 (3-КНФ):

$$F_M = C_1^3 \wedge C_2^3 \wedge ... \wedge C_M^3$$

.

Трехлитеральный дизъюнкт $C_j^3(x_k^p, x_\ell^q, x_m^r)$ – это дизъюнкция трех литералов переменных $X_\imath$ из множества $\mathbf{X}$. Пример дизъюнкта с конкретными значениями верхних индексов литералов: $C_j^3(x_k^0, x_\ell^1, x_m^0) = x_k^0 \vee x_\ell^1 \vee x_m^0 = X_k \vee \neg X_\ell \vee X_m$.

Для проведения вычислительного эксперимента необходимо иметь возможность, при заданном числе N переменных генерировать желаемое количества V формул с заданным свойством - невыполнимостью. В настоящем исследовании использован широко известный генератор 3-КНФ формул [1-3], представленный в одной из эквивалентных формулировок. Для построения каждой из формул используется своя последовательность $\mathbf{S}_v$ случайных натуральных чисел.

Формула, не имеющая дизъюнктов выполнима. Искомая невыполнимая формула строится так, что, пока она остается выполнимой, к ранее построенным дизъюнктам добавляется новый.

Генератор: Пусть построены j дизъюнктов. При построении очередного дизъюнкта C_{j+1}^3 формулы F_M из множества литералов $\mathbf{A}$ случайным образом выбирается элемент x_k^p, который полагается первым литералом дизъюнкта. Второй литерал x_ℓ^q случайным образом назначается из множества $\mathbf{A}$ - $\{x_k^0, x_k^1\}$, содержащего $2N-2$ элементов. Третий литерал дизъюнкта случайным образом выбирается из множества $\mathbf{A}$ - $\{x_k^0, x_k^1, x_\ell^0, x_\ell^1\}$, содержащего $2N-4$ элементов. Выбранный элемент x_m^r становится третьим литералом дизъюнкта $C_{j+1}^3(x_k^p, x_\ell^q, x_m^r)$.

Случайный выбор литералов осуществляется с использованием случайных чисел из последовательности $\mathbf{S}_v$.

Такой генератор исключает возможность построения дизъюнкта, содержащего более одного литерала одной переменной.

Если в результате добавления очередного дизъюнкта $C_{M_v}^3$ получается невыполнимая формула, генерация потока дизъюнктов завершается.

Замечательным свойством генератора является соблюдаемая с высокой точностью и подтвержденая многочисленными исследованиями [1-10] линейная зависимость $\mathfrak{M}$ от N:

$$\mathfrak{M}(N) = 4,23N + 7,18.$$

Если генерировать формулы содержащие по $\lfloor \mathfrak{M}(N) \rfloor$ дизъюнктов от N переменных то математическое ожидание вероятности выполнимости формул будет $\geqslant 0,5$. Если генерировать формулы содержащие по $\lceil \mathfrak{M}(N) \rceil$ дизъюнктов от N переменных, то математическим ожиданием вероятности невыполнимости будет величина большая, либо равная 0,5.

3. Характеристика алгоритма для задачи выполнимости

Полные (способные проводить доказательство невыполнимости) алгоритмы строятся на основе DP и DPLL процедур. В основе DP процедуры лежит возможность уменьшения числа переменных в формуле за счет выполнения всевозможных резолюций относительно этой переменной. Такой алгоритм предполагает построение очень большого числа резольвент и из-за использования огромных объемов памяти практически не используется.

DPLL процедура предполагает построение дерева разбора. Это построение основано на поочередном частичном присваивании конкретных значений переменным формулы.

Стандартно задача доказательства невыполнимости ставится как задача вывода пустого дизъюнкта. Однако она можем быть поставлена и как задача вывода литерально-полного дизъюнкта от трех переменных. Литерально-полным дизъюнктом является восьмерка дизъюнктов, содержащих все возможные комбинации литералов от трех переменных.

Отказ от непосредственного вывода пустого дизъюнкта увеличивает полиномиальную компоненту в оценке алгоритмической сложности, но при этом уменьшает потенциально экспоненциальную ее составляющую $E(N)$.

Задачу оценки алгоритмической сложности задачи выполнимости переформулируем как задачу оценки максимальной сложности вывода одного из дизъюнктов. Полиномиальность/экспоненциальность новой задачи соответствует полиномиальности/экспоненциальности оценки трудоемкости для задачи выполнимости.

В эксперименте использован алгоритм анализа, содержащий в себе черты как DP процедуры, использующей резолюции для построения новых дизъюнктов, так и характерное для DPLL построение дерева анализа с использованием частичного присвоения значений булевым переменным.

Каждое частичное присвоение, осуществляемое в условиях выбора, порождает новую ветвь в древовидной структуре. Используемый алгоритм после каждого частичного присвоения, осуществляемого однозначно или в условиях выбора, запускает резолюционную процедуру построения новых дизъюнктов, содержащих не более трех литералов. Эта процедура полиномиальна, поскольку ограниченная длина дизъюнктов ограничивает их число. Наличие такой процедуры увеличивает полиномиальную компоненту в оценке алгоритмической сложности, но при этом уменьшает количество ветвлений при построении древовидной структуры.

Построение и анализ каждой из ветвей древовидной структуры является процедурой полиномиальной сложности. Полиномиальность/экспоненциальность оценки сложности непосредственно зависит от полиномиальности/экспоненциальности числа ветвей в древовидной структуре анализа формулы.

Промежуточной целью используемого алгоритма является минимально сложный вывод новых трехлитеральных дизъюнктов. Невыполнимость формулы доказана, когда на какой-либо тройке переменных получается литерально-полный дизъюнкт.

Количество ветвей максимально сложного вывода дизъюнкта определяет полиномиальность, субэкспоненциальность или экспоненциальность искомой оценки алгоритмической сложности.

4. Результаты вычислительного эксперимента

Для каждого значения числа N переменных по изложенной выше методике построено по 201 невыполнимых формул $V = 201$.

В процессе доказательства невыполнимости формулы F_v методом резолюций выводятся новые трехлитеральные дизъюнкты до тех пор пока по каким либо трем переменным не получится полнолитеральный-дизъюнкт.

В первую очередь алгоритм старается вывести дизъюнкты, вывод которых (на основании процедур эвристического выбора) кажется наиболее простым. При этом фиксируется количество ветвлений B_v, имевших место при самом сложном выводе одного из дизъюнктов формулы F_v.

Из V значений B_v выбирается медианное значение $ME(N, V)$

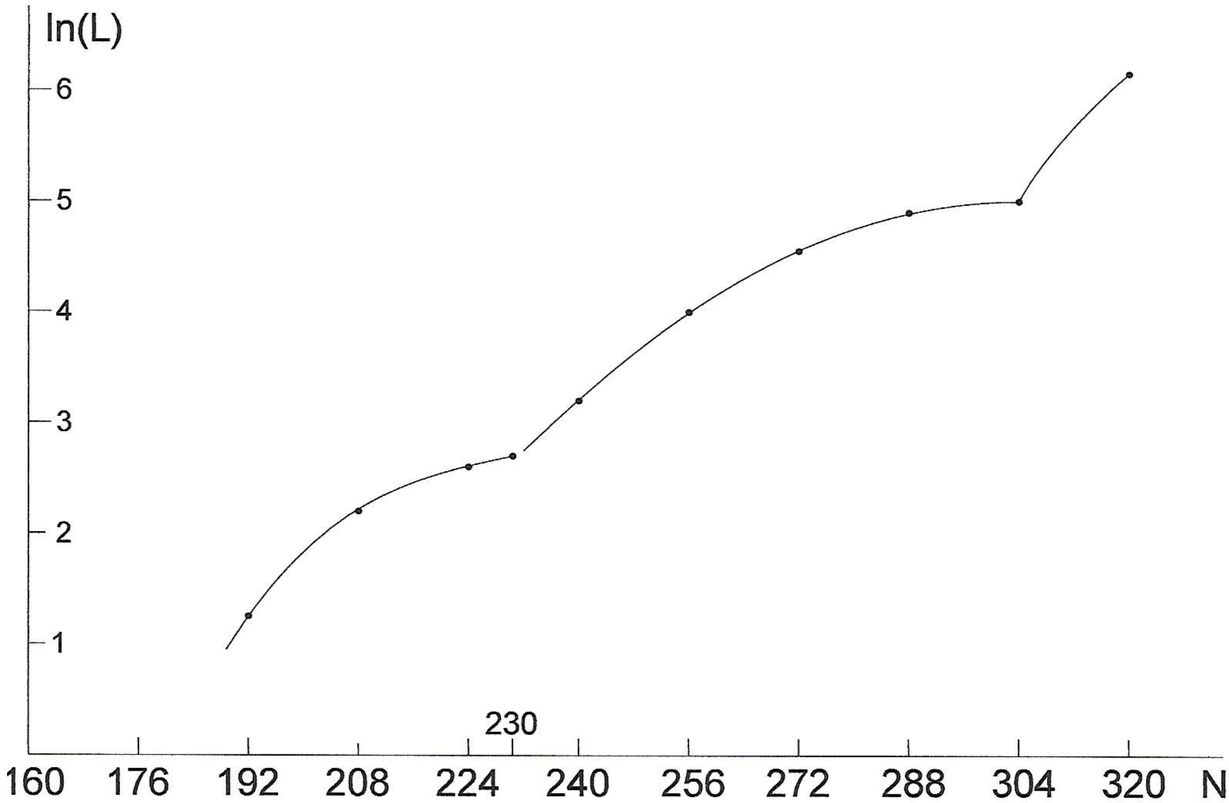

Рис. 1. Логарифмическая зависимость медианной сложности построения новых дизъюнктов от числа переменных в 3-КНФ формулах

На рис. 1 представлены основные результаты проведенного вычислительного эксперимента. По горизонтальной оси отложено число N переменных в 3-КНФ формулах. По вертикальной оси – логарифм от медианного (по 201 формул) значения максимальной для каждой формулы сложности B_v резолюционного вывода нового дизъюнкта. В проведенном эксперименте вплоть до 180 переменных потенциально экспоненциальная компонента трудоемкости $E(N)$ не проявляется. Если при продолжении эксперимента для больших значений числа переменных выяснится, что общей ассимптотой графика является наклонная прямая, это будет свидетельствовать в пользу экспоненциальной зависимости от N сложности доказательства невыполнимости 3-КНФ формул использованным алгоритмом.

5. Заключение

Полученные результаты свидетельствуют о том, что до 304 переменных характеристика алгоритмической сложности представляется полиномиальной. Затем имеет место значительный скачек сложности. Поэтому исключительный интерес представляет продолжение эксперимента при большем числе переменных в генерируемых 3-КНФ формулах.

Список литературы

1. Cook S., Mitchell D. Finding hard instances of the satisfiability problem: a survey // DIMACS Serries in Discr. Math. and Theoretical Comp. Sci. 1997. Vol. 5. P. 1-17.
2. Beame P., Karp R., Pitassi T., Saks M. On the Complexity of Unsatisfiability Proofs for Random k-CNF Formulas // 30th STOC. Dallas, TX, May 1998. P. 561-571.
3. Goldberg A. On the complexity of the satisfiability problem. 1979. 96 p.
4. Mitchell D., Selman B., Levesque H. Hard and easy distribution of SAT problem // Proceeding of the Tenth National Conference on Artificial Intelligence (AAAI-92). San Jose, CA. 1997. P. 459-465.
5. Crawford J., Auton I. Experimental Results on the Crossover Point in Satisfiability Problems // Proceeding of AAAI-93. Washington, DC. 1993. P. 21-27.
6. Gomes C., Kautz H., Sabharwal A., Selman B. Satisfiability Solvers // Handbook of Knowledge Representa-tion. Elsevier, 2008. P. 88-134.
7. Heule M. Minimal unsatisfiable cores of random formulas // Proceeding of SAT competition 2013. Solver and benchmark description. P. 105.
8. Biere A., Heule M., Maaren H., Walsh T. Handbook of Satisfiability. IOS Press, 2009. 966 p.
9. Mohammad Tghi Hajighayi , Gregory Sorkin. The Satisfiability Threshold of Random 3-SAT is at Least 3.52 // 2003/10/13. arXiv preprint math /0310193.
10. Kaporis A.C., Kirousis L.M., Laias E.G. The Probabilistic Analysis of a Greedy Satisfiability Algorithm // Random Structures Algorithms. 2006. Vol. 28 (40). P. 444-480.