АЛГОРИТМ МИНИМИЗАЦИИ ФУНКЦИОНАЛА, АССОЦИИРОВАННОГО С ЗАДАЧЕЙ 3-SAT И ЕГО ПРАКТИЧЕСКИЕ ПРИМЕНЕНИЯ

В.И. Дулькейт, Р.Т. Файзуллин, И.Г. Хныкин
Омский государственный университет им. Ф.М.Достоевского

Аннотация

Одной из наиболее интересных задач дискретной математики является задача поиска решающего набора в задаче ВЫПОЛНИМОСТЬ. После классической работы Кука [5] усилия многих исследователей были направлены на построение эвристических, переборных алгоритмов решения КНФ. Перспективным направлением представляется и сведение КНФ к непрерывному аналогу, к задаче поиска точек глобального минимума ассоциированного функционала. В данной работе обосновывается выбор функционала специального вида и предлагается применить к решению системы нелинейных алгебраических уравнений, определяющих стационарные точки функционала, модифицированный метод последовательных приближений. В работе также показано, что метод может быть с успехом применен к важным задачам криптографического анализа несимметричных шифров.

Введение

Пусть $L(x)$ – пропозициональная формула в конъюнктивной нормальной форме на множестве булевых переменных $x \in B^N\{0,1\}$. Задача ВЫПОЛНИМОСТЬ (SAT) заключается в том, что бы найти решающий набор $x_0 \in B^N\{0,1\}$, такой что $L(x_0) = ИСТИНА$ или доказать, что решающего набора не существует.

Рассмотрим переход от задачи ВЫПОЛНИМОСТЬ к задаче поиска глобального минимума функционала.

Пусть дана КНФ:

$L(x) = \bigcap_{i=1}^{M} c_i$, где c_i – дизъюнкты вида

$c_i = \bigcup_{j<N} q_{i,j}(x_j)$. Здесь $q_{i,j}(x_j) = x_j \; или \; \overline{x}_j$

Сделаем переход к эквивалентной ДНФ:

$L = \tilde{L}(x) = \bigcup_{i=1}^{M} \tilde{C}_i$, где $\tilde{C}_i$ – конъюнкты вида

$\tilde{C}_i = \bigcap_{j<N} \tilde{Q}_{i,j}(x_j)$. Здесь $\tilde{Q}_{i,j}(x_j) = \overline{q}_{i,j}(x_j)$

Рассмотрим функционал вида:

$$\min_{x \in E^N} F(x) = \sum_{i=1}^{M} C_i(x), \text{ где}$$

$$C_i(x) = \prod_{j=1}^{N} Q_{i,j}(x_j), \text{ где} \qquad (1)$$

$$Q_{i,j}(x_j) = \begin{cases} (1-x_j)^2 & , если \quad x_j \in C_i(x) \\ x_j^{\,2} & , если \quad \overline{x_j} \in C_i(x) \\ 1 & , иначе \end{cases}$$

Суммирование ведется по всем М конъюнктам ДНФ, эквивалентной исходной КНФ. Соответствие между булевыми и вещественными переменными следующее: ЛОЖЬ→0, ИСТИНА→1.

Переход от булевой формуле к вещественной основан на использовании соответствия:

$$\begin{cases} y_i \vee y_j \to x_i + x_j \\ y_i \wedge y_j \to x_i^2 x_j^2 \quad , \text{где } \{y_i \in B, x_i \in R\} \\ \overline{y}_i \to (1 - x_i) \end{cases}$$

Легко заметить, что $\min_{x \in E^N} F(x) = 0$ соответствует достижению значения ИСТИНА на исходной КНФ.

Без потери общности можно рассмотреть 3-ДНФ, эквивалентную исходной КНФ:

$J(x) = \sum_{\xi} z_i^2 z_j^2 z_k^2$, где

$$z_i = \begin{cases} 1 - x_i & , если \quad x_i \in c_i(x) \\ x_i & , если \quad \overline{x_i} \in c_i(x) \end{cases} \qquad (2)$$

Здесь $c_i(x) - i$ триплет

Дифференцируя функционал по всем переменным x_i, получаем систему уравнений:

$$\sum_{\xi \in \Xi} z_j^2 z_k^2 x_i = \sum_{\xi \in \Lambda} z_j^2 z_k^2, \quad i = 1,2,..P, \quad \text{где}$$

$$\Xi = \{\xi, i \in \xi : x_i \in c_i(x)\}$$

$$\Lambda = \{\xi, i \in \xi : \overline{x_i} \in c_i(x)\} \text{ или} \qquad (3)$$

$$A_i(x_1,..,x_{i-1},x_{i+1},..x_n) \cdot x_i = B_i(x_1,..,x_{i-1},x_{i+1},..x_n), \quad i = 1,..P$$

Коэффициенты A_i и B_i связаны соотношением:

$$A_i(x_1,..,x_{i-1},x_{i+1},..x_n) \geq B_i(x_1,..,x_{i-1},x_{i+1},..x_n)$$

Поясним выбор представления исходной КНФ именно в виде эквивалентной 3-ДНФ. Дифференцируя функционал $F(x)$ (1) по всем переменным x_i, получаем систему уравнений аналогичную (3), но количество вкладов в A_i и B_i определятся длиной скобок. Любая процедура решения этой системы при произвольной длине скобок будет естественным образом приводить к большим ошибкам округления. Ограничивая число переменных в скобках, мы исключаем эту техническую трудность.

Рассмотрим систему (3), как нелинейное операторное уравнение:

$$\Phi(x) = 0 \tag{4}$$

Как показано в [5] применение метода Ньютона к решению данного уравнения неэффективно, т.к. решение принадлежит ядру производного оператора. Как альтернатива был предложен метод последовательных приближений с «инерцией»[2]:

$$\Big(\sum_{p=0}^{K}\sum_{\xi\in\Xi}\alpha_p x_i(t-p)^2 x_j(t-p)^2\Big)x_k(t+1) = \sum_{\xi\in\Lambda} x_j^2(t)x_k^2(t) \sim A\cdot x_k(t+1) = B \tag{5}$$

$$\sum_{p=1}^{K}\alpha_p = 1 \qquad ,где \quad \alpha_p \in R[0,1]$$

Имеется ввиду то, что итерации происходят для вещественных чисел, а итоговый или промежуточный вектор проектируется на $\mathrm{B}^{\mathrm{N}}\{0,1\}$, и уже на булевом векторе проверяется SAT. Ниже мы опишем различные модификации и гибридизации метода последовательных приближений с «инерцией» в применении к решению задачи K-SAT, и покажем способы повышения эффективности алгоритма.

Гибридизация алгоритма

Основная процедура состоит из последовательных итераций, которые совмещают метод последовательных приближений и сдвиг по антиградиенту, т.к. правая часть (3) это не что иное, как антиградиент исходного функционала.

Итерация состоит из двух блоков. Первый блок, определяется формулой (5), используется схема Зейделя. Суть схемы Зейделя в том, что при нахождении очередного $x_i(t+1)$ на $(t+1)$-й итерации, это значение подставляется вместо $x_i(t)$. Необходимо отметить, что реализация алгоритма допускает использование схемы Якоби, когда найденные $x_i(t+1)$ не используются в текущей итерации. Тесты показали, что схема Зейделя более устойчива в применении к решаемой задаче. Именно, после каждой итерации по схеме Зейделя значение функционала монотонно уменьшается, чего нельзя сказать о схеме Якоби. Кроме того, во всех случаях схема Зейделя быстрее приводила к решению. Отметим, что данное обстоятельство препятствует напрашивающейся простой схеме распараллеливания процесса решения с разделением данных.

Второй блок – реализация сдвига по градиенту. Рассмотрим (4). Пусть $\bar{x}(t)$ является решением, тогда $\Phi(\bar{x}(t)) = 0$. Уравнение (5) переписывается в виде $A(\bar{x}(t))\cdot\bar{x}(t) - B(\bar{x}(t)) = 0$. Это необходимое условие, которому должен удовлетворять вектор решения. Если текущее t–е приближение $\bar{x}(t)$ не является решением, то $A_i(\bar{x}(t))\cdot x_i(t) - B_i(\bar{x}(t)) = p_i \neq 0$. Для итеративной формулы: $A_i(\bar{x}(t))\cdot x_i(t+1) - B_i(\bar{x}(t)) = p_i$. Следовательно, что бы удовлетворить необходимому условию, необходимо перейти к вектору:

$$\bar{x}_i(t+1) = \bar{x}_i(t) + p_i / A_i \tag{6}$$

Очевидно, что после реализации (6) возможна ситуация, когда $\bar{x}_i(t+1) \notin R[0,1]$. В этом случае необходимо штрафным способом ограничивать $\bar{x}_i(t+1)$, иначе метод начинает экспоненциально расходиться. Особенно это проявляется на K-SAT формулах при K>4. При приближении к решению скорость сходимости может сильно уменьшаться, т.е. алгоритм формирует цикл лежащий на некотором плато (поверхность, определяемая функционалом) и траектория, образованная последовательными приближениями более не выходит за пределы этого плато. Чтобы сойти с плато и продолжить сходимость к решению применяется т.н. метод смены траектории.

Метод смены траектории заключается в поиске нового вектора приближения, который бы обладал свойствами не худшими, чем текущий вектор приближения, но позволял бы продолжить поиск решения. Суть метода нахождения такого вектора в следующем.

Рассмотрим 3-КНФ, эквивалентную исходной 3-ДНФ (1):

$$K(x) = \prod_{\xi}(z_i + z_j + z_k), \text{ где}$$

$$z_i = \begin{cases} x_i & ,если \quad x_i \in c_i(x) \\ 1 - x_i & ,если \quad \overline{x_i} \in c_i(x) \end{cases} \tag{7}$$

Здесь $c_i(x) - i$ триплет.

$$K(x) = 1 \Leftrightarrow (z_i + z_j + z_k) \neq 0, \forall\xi$$

Для данного приближения $\bar{x}$, рассмотрим множество переменных:

$$E0 = \left\{ x_i \,\middle|\, \exists \text{ триплет } c_i : x_i \text{или} \neg x_i \in c_i \;\&\; c_i(\bar{x}) = 0 \right\}$$

С вероятностью m_p поменяем значения x_i на про-

тивоположные. При этом, вероятность того, что другие триплеты станут невыполнимыми не высока. Экспериментально установлено, что полученный вектор x_0 обладает свойствами не худшими, чем $\overline{x}$ (количество невыполнимых триплетов до и после операции примерно одинаково). Используя данный вектор x_0 в качестве нового начального приближения, алгоритм очень быстро (в большинстве случаев за 5-10 итераций) находит следующее приближение, на котором функционал $F(x)$ достигает значения не хуже, чем на векторе $\overline{x}$. При этом, очень часто, удается проскочить плато, но при дальнейшем движении по новой траектории метод может зациклиться на другом плато. Тогда метод смены траектории повторяется.

Дополнительно рассмотрим множество:

$$E1 = \left\{ x_i \,\middle|\, \exists \text{триплет } c_i : x_i \text{или} \neg x_i \in c_i \;\&\; c_i(\overline{x}) = 1 \right\}$$

Введем вероятности m_p0, m_p1. С вероятностью m_p0 при смене траектории будем использовать множество $E0$. С вероятностью m_p1 – множество $E1$. Вероятность m_p0 влияет на величину изменения вектора и при этом количество невыполнимых триплетов не увеличивается. Вероятность m_p1 влияет на качественное изменение вектора, количество невыполнимых триплетов может увеличиться. В принципе, чем выше m_p1, тем меньшую роль играют рестарты. Метод смены траектории применяется при достижении условия $\left|F(\overline{x}_2) - F(\overline{x}_1)\right| < \varepsilon_2$.

Преобразование исходной КНФ методом резолюции

Преобразование позволяет получить КНФ с меньшим количеством дизъюнктов и литералов, эквивалентную исходной.

«Резольвента» – дизъюнкция конъюнктов, отличающихся знаком по единственной переменной. Все возможные резольвенты добавляются к КНФ и используются для вычисления других резольвент.

Дублирующие конъюнкты и тавтологии удаляются, и используется сокращенная процедура с глубиной рекурсии 1. Вычислительная сложность процедуры $O(n \cdot \log n)$. Метод резолюции в применении к КНФ, ассоциированных с задачами факторизации и дискретного логарифмирования (см. ниже) позволяет уменьшить исходное число конъюнктов до 50%.

Подробно о методе резолюций можно найти в [4]

Результаты численных экспериментов

После каждой модификации проводилось тестирование алгоритма для определения эффективности проделанных изменений. При тестировании использовалось несколько типов примеров: тесты с соревнований решателей SAT 2005 года www.lri.fr/~simon/, тесты библиотеки SATLib www.cs.ubc.ca/~hoos/, тесты, сформированные для задач факторизации и дискретного логарифмирования, тесты для КНФ больших размерностей, сформированные случайным образом.

Результаты - метод последовательных приближений с инерцией

Основной результат вычислительных экспериментов относительно модифицированного метода последовательных приближений, проводившихся для случайного наполнения наборов скобок SAT и 3-SAT, представлен в [3]. При соотношении $\theta = \frac{N}{M} \le 0.5$, где N это число переменных, M число скобок в 3-SAT (эксперименты проводились вплоть до $N \approx 10^6$, $M \approx 2 \cdot 10^6$), итерационная процедура всегда сходится к решению. Но при увеличении θ скорость сходимости резко падает и, хотя большинство компонент решения $\overline{x}$ формируется верно, и быстро, оставшаяся часть оказывается практически недостижимой.

Результаты - метод последовательных приближений с инерцией (+ сдвиг по градиенту)

Оказалось, что сдвиг по градиенту хорошо сокращает погрешности и ускоряет сходимость алгоритма. Вычислительные эксперименты со случайными формулами показали заметное уменьшение времени решения тестов. Число решенных примеров увеличилось примерно на 20%. Применение данного приема позволило достаточно эффективно решать тесты uf20-91 (из 1000 тестов решены 703). Однако некоторые тесты решались только после задания определенного начального приближения, что говорит о необходимости рестартов. На примерах uf250-1065 алгоритм показал результат 6% от стандартных трудных тестов (предыдущая версия алгоритма - 1%). Тесты SAT-2005 (OKGenerator10000-42000 – 10000 переменных, 42000 скобок) использовались в сокращенной форме. Максимально удалось решить подформулы из 36000 скобок (предыдущая версия алгоритма – 35000 скобок). На подформулах из более чем 36000 скобок метод зацикливается на некотором плато. Это говорит о необходимости смены траектории.

Результаты - метод последовательных приближений с инерцией (+ метод смены траектории)

Метод смены траектории существенно увеличил число решаемых примеров. Результаты представлены в таблице 1.

Увеличение разрядности вычислений

Была исследована сходимость алгоритма при увеличении разрядности вычислений. Испытания с типами DOUBLE и FLOAT показали преимущество

вычислений с двойной точностью. При переходе на тип DOUBLE количество решенных примеров увеличивается на 10%, скорость сходимости в среднем также увеличивается. Дальнейшее увеличение разрядности к значимому эффекту не приводит.

Таблица 1. Результаты тестирования алгоритма + метод смены траектории

Наименование теста	Количество литералов (N)	Количество дизъюнктов (M)	Число тестов	% решенных тестов	Максимальное число итераций
Backbone-minimal Sub-instances (формулы с минимальным хребтом), 3-SAT					
RTI	100	429	500	98,6	19988
BMS	100	<429	500	79,8	29831
Controlled Backbone Size Instances (b – размер хребта), 3-SAT					
CBS_b10	100	403	1000	100	38972
CBS_b10	100	449	1000	100	38880
CBS_b90	100	449	1000	98	29738
Uniform Random 3-SAT (UF)					
uf20-91	20	91	1000	100	448
uf250-1065	250	1065	100	98	9731
SAT-encoded "Flat" Graph Colouring Problems					
flat30-60	90	300	100	100	4317

Возможные практические применения

В последнее время наблюдается повышенный интерес к проблеме кодирования криптографических алгоритмов в терминах задачи выполнимости (SAT). В работе [1] эскизно иллюстрируется подход, позволяющий в принципе свести задачу факторизации к SAT. Целью данного пункта статьи является построение алгоритмов генерации эквивалентных, но различных КНФ и последующей минимизации ассоциированного функционала, для задачи факторизации, задачи дискретного логарифмирования, задачи дискретного логарифмирования на эллиптической кривой.

Результаты работы алгоритма для задачи факторизации

Результаты приведены в табл. 2.

Таблица 2. Результаты тестирования полного алгоритма для задачи факторизации.

1*	2*	3*	4*	5*	6*
20	254	4979	0.1 с	0.1 с	0.1 с
32	801	17867	2 м	>1 ч	1.8м
40	990	22333	7 м	>1 ч	12 м
44	1199	27291	36 м	>1 ч	>1 ч
48	1428	32741	3,5 ч	>10ч	>10ч
56	1946	45141	36 м	>10ч	>10ч
60	2235	52079	10,2ч	>20ч	>20ч
68	2873	67455	79 ч	>100ч	>100ч
72	3222	75881	168ч	>200ч	>200ч

1*- Число бит в факторизуемом числе
2*- Количество литералов
3*- Количество дизъюнктов
4*- Время решения методом последовательных приближений
5*- Время решения алгоритмом RANOV (победитель 2005 г.)
6*- Время решения алгоритмом SATz (один из лучших переборных алгоритмов)
Знак '>' означает, что за указанное время решение найдено не было

Для группы задач длины до 72 бит факторизуемого числа были получены точные решения. При этом эффективность предложенного метода превосходит известные нам алгоритмы.

Другой интересный результат в том, что уже после первых нескольких сотен итераций метод находит более 59% верных бит решения. Трудность заключается в определении, какие именно биты были определены верно. При этом при росте числа бит исходного факторизуемого числа, происходит рост процентного отношения числа верных бит для соотношения, определяющего факторизуемое число (см. Рис. 1).

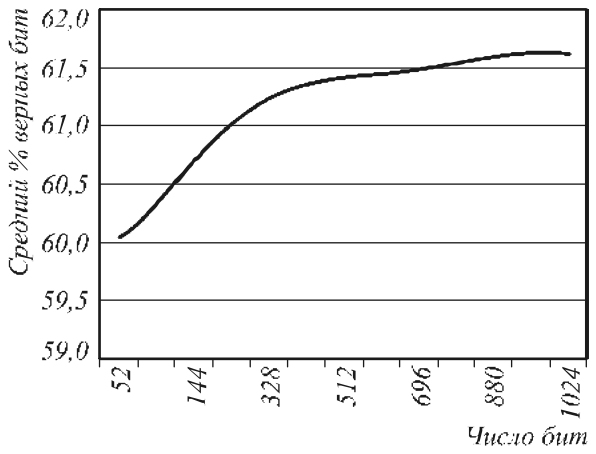

Рис.1. Рост процентного отношения числа верных бит к числу бит факторизуемого числа.

Для тестирования использовалось 50 различных тестов для каждой размерности. В качестве сомножителей выбирались числа, удовлетворяющие всем тестам, гарантирующим криптостойкость RSA. На

рисунке представлены усредненные данные. Обратим внимание на то, что вне зависимости от размерности задачи для каждого теста проводилось всего 1000 итераций. Исходя из таблицы 2 можно сделать предположение о том, что размерность 56 бит является «слабой» для алгоритма минимизации.

Сведение к КНФ задачи факторизации

Рассмотрим непосредственно алгоритм сведения к КНФ задачи факторизации. Требуется для заданного числа n получить КНФ, решающий набор которой существует тогда, и только тогда когда n составное число. Кроме того, решающий набор должен содержать все биты двоичного представления нетривиальных делителей n. Без потери общности, рассмотрим классический алгоритм умножения «столбиком». Будем отождествлять биты сомножителей и результата с литералами (свободными логическими переменными). Результат умножения первого сомножителя на i-ый бит второго можно представить в виде вектора $\overline{P_i}$. Именно суммирование всех этих векторов представляет основную сложность. Поэтому предлагается выполнять эту операцию последовательно с сохранением результата в промежуточных векторах $\overline{S_k}$.

Весь процесс вычисления можно разбить на три этапа относительно операции сложения:

1) Сложение векторов составленных из произведений двух литералов. Выполняется один раз. В результате этой операции будет заполнен вспомогательный вектор сумм $\overline{S_1}$ и вычислено два младших бита результата. Условно данный этап можно записать так: $\overline{P_1} + \overline{P_2} = (\overline{S_2}, r_2, r_1)$.
2) Суммирование вектора $\overline{S_k}$ с вектором произведений. Выполняется N-3 раз. В результате заполняется массив $\overline{S_{k+1}}$ и вычисляется очередной бит результата

$$\overline{S_{i-2}} + \overline{P_i} = \left(\overline{S_{i-1}}, r_i\right), i = 3 \ldots N-1$$

Последнее суммирование вектора $\overline{S_k}$ с вектором произведений. Выполняется один раз. В результате вычисляются оставшиеся биты результата.

$$\overline{S_{N-2}} + \overline{P_N} = \left(r_{2N} \ldots r_N\right)$$

Кодирование основных операций

Теперь перейдём к рассмотрению идеи генерации КНФ. Простейший случай – приравнивание одного литерала другому: $x = y$. Данное равенство будет справедливо тогда и только тогда, когда истинна формула $\left(\overline{x} \vee y\right)\left(x \vee \overline{y}\right)$.

Другим часто встречающимся выражением является:

$$x = A \oplus B \oplus C \tag{10}$$

Поступая аналогичным образом, получаем эквивалентную формулу:

$$\begin{gathered}\left(\overline{x} \vee \left(A \oplus B \oplus C\right)\right)\left(x \vee \overline{\left(A \oplus B \oplus C\right)}\right) = \\ \left(Ax \oplus Bx \oplus Cx \oplus \overline{x}\right)\left(A\overline{x} \oplus B\overline{x} \oplus C\overline{x}\right)\end{gathered} \tag{11}$$

Для представления правой части в виде КНФ воспользуемся леммами 1, 2.

Лемма 1.

$\bigoplus_{i=1}^{N} x_i = \prod_{\{\delta_i\} \in M_N} \left(x_1^{\delta_1} \vee x_2^{\delta_{21}} \vee \cdots \vee x_N^{\delta_N}\right)$, где в левой части сумма по модулю 2, M_N – множество двоичных векторов длины N, содержащих чётное число нулей. Операция «возведения в степень» имеет стандартный для булевой алгебры смысл:

$$x^{\delta} = \begin{cases} \overline{x}, & \delta = 0 \\ x, & \delta = 1 \end{cases}$$

Лемма 2.

$$\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \prod_{j=1}^{L} y_j^{\sigma_j} =$$

$$\prod_{\{\pi_k\} \in 2^L / \{0,0,\ldots,0\}} \left(\bigvee_{i=1}^{N} x_i^{\delta_i} \vee \bigvee_{j=1}^{L} \left(y_j^{\sigma_j}\right)^{\pi_k} \right)$$

После применения леммы 1 будут получены конъюнкты следующего вида:

$\left(d \vee \overline{a\overline{b}c} \vee x\overline{y}c\right)$, т.е. можно выделить 3 вида дизъюнктов внутри каждого конъюнкта:

1) Одиночные литералы (то к чему следует стремиться: в правильной КНФ все дизъюнкту должны быть одиночными литералами).
2) Дизъюнкты вида $\overline{\prod x_i^{\delta_i}}$, которые по правилу де Моргана можно легко свести к одиночным литералам.
3) Дизъюнкты вида $\prod x_i^{\delta_i}$, наиболее трудный случай, сведение скобок с такими дизъюнктами к КНФ иллюстрируется леммой 2.

Отдельного рассмотрения заслуживает операция вычисления переноса в следующий разряд при суммировании трёх слагаемых. Перенос может быть вычислен через соответствующую сумму:

$$c = carry(x, y, z, sum) = \overline{\left(sum \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}}\right)}$$

Выполнение данного равенства эквивалентно истинности следующей формулы:

$$\left(\bar{c} \vee \overline{\left(sum \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}}\right)}\right) \cdot \left(c \vee \left(sum \oplus xyz \oplus \overline{\bar{x}\bar{y}\bar{z}}\right)\right).$$

Приведённое выражение можно преобразовать положив: $x = \bar{c}$, $A = sum$, $B = xyz$, $C = \overline{\bar{x}\bar{y}\bar{z}}$. И далее описанной выше процедурой можно построить соответствующую КНФ. Трудоемкость полученного алгоритма оценивается, как $O(n^2)$ в зависимости от количества бит исходного числа. Для факторизации числа, представляемого двоичным вектором длиной 1024 бит получились КНФ с 500 000 переменными 12000 000 скобок. Отметим, что результат о превышении числа верных бит после нескольких тысяч итераций алгоритма (5) относится именно к числу переменных, например, из 500 000 переменных мы получаем в среднем 300 000 верных, что в силу очевидных соотношений между битами переноса по строке, отвечающей нулевому биту, может существенно облегчить решение основной задачи.

Сведение к КНФ других задач криптоанализа

Были построены аналогичные алгоритмы сведения для задач дискретного логарифмирования и дискретного логарифмирования на эллиптической кривой. Предложен алгоритм генерации множества эквивалентных КНФ для задачи факторизации, учитывающий неделимость на малые простые числа. Последнее обстоятельство позволяет строить параллельные версии приведенных выше алгоритмов и методикой «голосования бит» определять верные биты с большой вероятностью.

Заключение

Разработанный метод не уступает известным методам решения SAT на многих группах тестовых примеров и превосходит их на тестах задачи факторизации больших размерностей. Показан рост относительного числа верно найденных бит для задачи факторизации с ростом размерности задачи, так для рабочего числа бит 1024 среднее значение верных бит равно 61.75%, что больше стартового значения в 59.5% для 50 бит. Кроме того, показано наличие «слабых» размерностей, для которых метод является эффективным. Аналогичные результаты получены и для задачи дискретного логарифмирования, но основным препятствием здесь является высокая размерность получающихся задач, так для 1024 бит число переменных в функционале оценивается величиной 10 000 000 000, а число дизъюнктов на порядок больше. Результаты точного решения КНФ, эквивалентных задаче дискретного логарифмирования приведены в таблице 3.

Таблица 3. Результаты тестирования полного алгоритма для задачи дискретного логарифмирования.

1*	2*	3*	4*	5*	6*
18	28224	448018	63.57	97.23	81.16
20	38840	623239	108.20	>1800	>1800
22	51832	839032	182.73	>1800	>1800
24	67440	1099630	277.46	>1800	>1800
26	85904	1409250	417.71	>1800	>1800

1*- Размерность, бит
2*- Количество литералов
3*- Количество дизъюнктов
4*- Время решения методом последовательных приближений, сек
5*- Время решения алгоритмом RANOV, сек (победитель 2005 г.)
6*- Время решения алгоритмом SATz, сек (один из лучших переборных алгоритмов)
Знак '>' означает, что за указанное время решение найдено не было

Литература

1. **Беспалов Д.В.**, О логических выражениях для задачи 2-ФАКТОРИЗАЦИЯ // Беспалов Д.В. Семёнов А.А.Вычислительные технологии. – 2002. – Т.7 – Ч.2.
2. **Файзуллин Р.Т.** О решении нелинейных алгебраических систем гидравлики // Сибирский журнал индустриальной математики. -1999.-№2. –С. 176-184.
3. **Файзуллин Р.Т.,** Алгоритм минимизации функционала, ассоциированного с задачей 3-SAT и его практические применения, // Файзуллин Р.Т., Хныкин И.Г., Дулькейт В.И., Салаев Е.В.- г.Челябинск, 2007.
4. **Хныкин И.Г.,** Модификация КНФ, эквивалентных задачам криптоанализа асимметричных шифров методом резолюции // ИТМУ № 8, 2007.
5. **Cook S.A.** The Complexity of Theorem Proving Procedures. Proceedings Third Annual ACM Symposium on Theory of Computing, May 1971.

ALGORITHM FOR MINIMIZATION OF FUNCTIONAL ASSOCIATED WITH 3-SAT PROBLEM AND ITS PRACTICAL USAGE

V.I. Dulkeyt[1], R.T. Faizullin[1], I.G. Khnykin[1]
[1] Omsk F.M. Dostoevsky State University

Abstract

One of the most challenging issues in discrete mathematics is a problem of finding a decisive set in SATISFIABILITY application. After Cook's classical paper [5], efforts of many researchers were aimed at build-up of heuristic direct-search algorithms to solve the satisfaction of a Conjuctive Normal Form (CNF). A perspective trend is the reduction of CNF to a continuous analogue, to searching absolute minimum of the associated function. In this paper we demonstrate feasibility of choice of a special function form and propose application of the modified method of successive approximations to solve a set of nonlinear algebraic equations, which define stationary points of the function. The paper also shows that the method can be successfully applied to solve critical issues of the cryptoanalysis of nonsymmetric codes.

Keywords: undecidable problems, computability theory, complexity classes.

Citation: Dulkeyt VI, Faizullin RT, Khnykin IG. Function minimization algorithm for 3-SAT and its practical applications [In Russian]. Computer Optics 2008; 32(1): 68-73.

References

[1] Bespalov DV, Semenov AA. About logical statements for 2-FACTORIZATION problem [In Russian]. Calculation Technologies 2002; 7(2).

[2] Faizullin,RT. On solution of nonlinear algebraic systems of hydraulics [In Russian]. Sib. Zh. Ind. Mat.1999; 2(2): 176-184.

[3] Faizullin RT, Khnykin IG, Dulkeyt VI, Salaev EV. Function minimization algorithm for 3-SAT and its practical applications [In Russian]. Chelyabinsk 2007.

[4] Khnykin IG. CNF modifications equivalent to problems of asymmetric cipher cryptoanalysis by a resolution technique [In Russian]. ITMU (Information Technologies for Modeling and Control) 2007; 8.

[5] Cook SA. The Complexity of Theorem Proving Procedures. Proceedings Third Annual ACM Symposium on Theory of Computing, May 1971.