**Поиск систем ортогональных латинских квадратов в проекте добровольных вычислений SAT@home**[1]

Заикин О.С., Кочемазов С.Е., Семенов А.А.
Институт динамики систем и теории управления СО РАН
zaikin.icc@gmail.com, veinamond@gmail.com, biclop.rambler@yandex.ru

**Ключевые слова**: латинские квадраты, SAT, добровольные распределенные вычисления, SAT@home

Вопросы существования систем ортогональных латинских квадратов интересуют математиков со времен Л. Эйлера. Латинский квадрат порядка $n$ – это квадратная $n \times n$ таблица, заполненная числами от 1 до $n$ таким образом, что в каждой строке и каждом столбце встречаются все числа от 1 до $n$. Пара латинских квадратов одинакового порядка называется ортогональной, если различны все упорядоченные пары чисел $(a,b)$, где $a$ – число в некоторой клетке первого латинского квадрата, а $b$ – число в клетке с тем же номером второго латинского квадрата. Если имеется набор из $m$ различных латинских квадратов, любая пара которых ортогональна, то говорят о системе из $m$ попарно ортогональных латинских квадратов. Одной из самых известных нерешенных задач, касающейся латинских квадратов, является следующая: определить, существует ли система из трех попарно ортогональных латинских квадратов порядка 10 [1].

Обширный класс задач верификации, криптографии, биоинформатики и поиска комбинаторных структур может быть эффективно сведен к задаче о булевой выполнимости (SAT [2]). Все известные алгоритмы решения SAT-задач экспоненциальны в худшем случае, т.к. SAT является NP-трудной задачей. Несмотря на это, современные SAT-решатели (в том числе параллельные и распределенные) успешно справляются с решением целого ряда задач из упомянутых выше областей.

Для использования SAT-подхода необходимо перейти от исходной постановки к булевому уравнению вида «КНФ=1» (КНФ – конъюнктивная нормальная форма). Такой переход называется пропозициональным кодированием исходной проблемы. Попытки применить SAT-подход к задаче поиска систем ортогональных латинских квадратов регулярно предпринимаются с середины 90-х годов XX века. Много полезной информации о применении SAT-решателей к поиску различных систем ортогональных латинских квадратов содержится в обзорной статье [3]. Автор этой статьи пробовал решить упомянутую выше задачу поиска тройки попарно ортогональных квадратов 10-го порядка с использованием SAT-решателя PSATO в течение более 10 лет в грид-системе из 40 рабочих станций (окончательный ответ так и не был получен).

Далее будет описана широко известная пропозициональная кодировка, встречающаяся во множестве различных работ (см. например, [3], [4]). Рассматриваем две матрицы $A = \|a_{ij}\|$ и $B = \|b_{ij}\|$, $i, j = \overline{1,\ldots,n}$. Содержимое каждой ячейки любой из матриц кодируется $n$ булевыми переменными. Для кодирования всей матрицы, таким образом, требуется $n^3$ булевых переменных. Будем использовать запись $x(i,j,k)$ и $y(i,j,k)$ для обозначения переменных, кодирующих элементы матриц $A$ и $B$, соответственно. При этом переменная $x(i,j,k)$ принимает значение «истина» тогда и только тогда, когда в ячейке, находящейся в строке с номером $i$ и столбце с номером $j$, стоит число $k$. Чтобы матрицы $A$ и $B$ представляли латинские квадраты необходимо наложить на соответствующие им переменные ряд условий, рассмотренные далее на примере матрицы $A$. Эти условия легко записываются в виде конъюнкций дизъюнктов.

[1]Работа выполнена при финансовой поддержке РФФИ (грант № 11-07-00377-а), и Совета по грантам Президента РФ для поддержки молодых ученых (стипендия СП-1855.2012.5).

- В каждой ячейке матрицы стоит ровно одно число от 1 до $n$:

$$\wedge_{i=1}^{n}\wedge_{j=1}^{n}\vee_{k=1}^{n}x(i,j,k)$$

$$\wedge_{i=1}^{n}\wedge_{j=1}^{n}\wedge_{k=1}^{n-1}\wedge_{r=k+1}^{n}(\neg x(i,j,k)\vee\neg x(i,j,r)).$$

- Каждое число от 1 до $n$ появляется в каждой строке ровно один раз

$$\wedge_{j=1}^{n}\wedge_{k=1}^{n}\vee_{i=1}^{n}x(i,j,k)$$

$$\wedge_{j=1}^{n}\wedge_{k=1}^{n}\wedge_{i=1}^{n-1}\wedge_{r=i+1}^{n}(\neg x(i,j,k)\vee\neg x(r,j,k)).$$

- Каждое число от 1 до $n$ появляется в каждом столбце ровно один раз

$$\wedge_{i=1}^{n}\wedge_{k=1}^{n}\vee_{j=1}^{n}x(i,j,k)$$

$$\wedge_{i=1}^{n}\wedge_{k=1}^{n}\wedge_{j=1}^{n-1}\wedge_{r=j+1}^{n}(\neg x(i,j,k)\vee\neg x(i,r,k)).$$

Аналогичным образом записываются условия, кодирующие матрицу $B$. После этого необходимо закодировать условие ортогональности. Например, это можно сделать следующим образом:

$$\wedge_{i=1}^{n}\wedge_{j=1}^{n}\wedge_{k=1}^{n}\wedge_{p=1}^{n}\wedge_{q=1}^{n}\wedge_{r=1}^{n}(\neg x(i,j,k)\vee\neg y(i,j,k)\vee\neg x(p,q,r)\vee\neg y(p,q,r)).$$

Мы использовали эту кодировку для поиска пар ортогональных латинских квадратов с дополнительным условием «диагональности». В таких парах в каждом квадрате как главная, так и побочная диагонали должны содержать все числа от 1 до $n$, где $n$ – это порядок этих квадратов. Известно, что такие пары порядка 10 существуют [5], однако в открытом доступе мы смогли найти только три пары из статьи [5]. Поэтому, на наш взгляд, было бы интересно найти новые пары диагональных ортогональных латинских квадратов порядка 10. Легко видеть, что для получения кодировки задачи поиска системы таких квадратов необходимо добавить в описанную выше кодировку дизъюнкты, соответствующие условию диагональности. Полученная в результате КНФ состоит из 2000 переменных и 434440 дизъюнктов, файл с КНФ занимает 10 мегабайт.

Задачи поиска систем ортогональных латинских квадратов при помощи SAT-подхода хорошо подходят для организации масштабных экспериментов в гридах, в частности, в проектах добровольных распределенных вычислений. Это объясняется тем, что SAT-задачи сами по себе допускают естественные стратегии крупноблочного распараллеливания. Авторами статьи в сотрудничестве с коллегами из ИППИ РАН был разработан проект SAT@home [6-8], функционирующий с сентября 2011 года, и предназначенный для решения трудных экземпляров SAT-задач из различных предметных областей. При создании проекта была использована открытая платформа BOINC [9]. По состоянию на 30 сентября 2013 года в проекте 2730 активно работающих ПК участников со всего мира, обеспечивающих среднюю производительность около 3.5 TFLOPs. Максимальная производительность (7.4 TFLOPs) была достигнута в начале сентября 2013 года. Схема решения SAT-задач с помощью проекта SAT@home представлена на рисунке 1. Предварительно осуществляется поиск «хорошей» декомпозиции SAT-задачи с помощью метода Монте-Карло на вычислительном кластере [7]. Необходимость использования кластера исходит из того, что на данном этапе используются интенсивные межпроцессорные обмены.

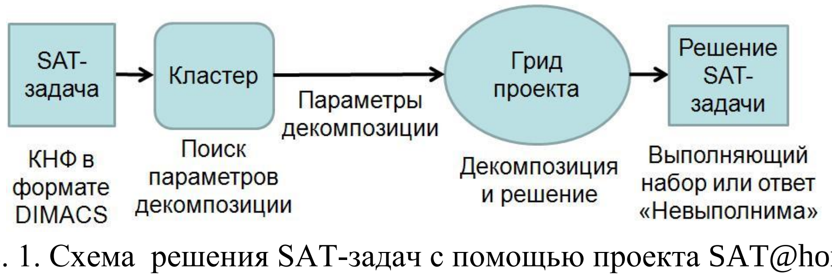


Рис. 1. Схема решения SAT-задач с помощью проекта SAT@home.

В мае 2012 года в проекте SAT@home был успешно завершен полугодовой эксперимент, направленный на решение 10 задач криптоанализа генератора ключевого потока A5/1, которые не решаются с помощью известных rainbow-таблиц [10]. В каждой задаче нужно было найти неизвестное начальное заполнение регистров генератора.

На следующем этапе в SAT@home был запущен поиск пар ортогональных диагональных латинских квадратов (ОДЛК) порядка 10. В качестве ядра клиентского приложения (которое запускается на ПК участников) использовался SAT-решатель MiniSat 2.2 [11], в который были внесены небольшие изменения, направленные на уменьшение используемой оперативной памяти. Первая строка первого квадрата была изначально зафиксирована и равнялась «0 1 2 3 4 5 6 7 8 9». Это было сделано потому, что любую пару ортогональных латинских квадратов путем перестановок, не нарушающих условия ортогональности и диагональности, можно преобразовать к паре, в которой первая строка первого квадрата будет именно такой. Выбор способа декомпозиции SAT-задач – нетривиальная проблема, в решении которой нужно использовать свойства исходных постановок [7]. Декомпозиция задачи поиска пар ОДЛК была осуществлена следующим образом. Варьировались значения первых 8 ячеек второй и третьей строки первого квадрата. Всего оказалось около 230 миллиардов возможных вариантов значений этих 16 ячеек, не нарушающих условие, что квадрат является диагональным латинским. Было решено сформировать для решения в SAT@home 20 миллионов подзадач из 230 миллиардов (т.е. всего около 0.0087 % описанного выше пространства поиска). В итоге каждая подзадача формировалась в результате подстановки в первый квадрат 8 первых ячеек второй и третьей строки (при фиксированной первой строке). Значения остальных 74 ячеек первого квадрата и всех 100 ячеек второго квадрата были неизвестны, их должен был найти SAT-решатель. На каждую подзадачу был установлен лимит в 2600 рестартов решателя MiniSat 2.2, что примерно соответствует 5 минутам работы одного ядра современного процессора. При достижении лимита вычисления прерывались. Каждое задание, которое получал участник SAT@home, состояло из 20 таких подзадач. На обработку 20 миллионов подзадач, сгенерированных для данного эксперимента, потребовалось около 9 месяцев работы проекта (с сентября 2012 года по май 2013 года). Вычисления практически для всех подзадач были прерваны при достижении лимита, но решение 17 подзадач закончилось успешно – в результате были найдены 17 новых пар ОДЛК порядка 10 (мы сравнивали их с тремя парами из статьи [5]). Все найденные пары выложены на сайте проекта [8] в разделе «Найденные решения». На рисунке 2 приведена первая пара ОДЛК порядка 10, найденная в проекте SAT@home.

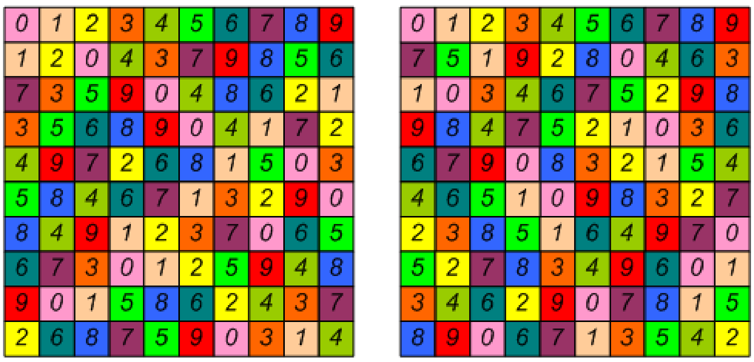


Рис. 2. Первая пара ортогональных диагональных латинских квадратов порядка 10, найденная в проекте SAT@home.

Изначально одной из приоритетных целей проекта SAT@home было решение знаменитой задачи о тройке попарно ортогональных латинских квадратов порядка 10. Однако, данная задача, как уже отмечалось выше, крайне трудна. Поэтому было принято решение сосредоточиться на ослабленных вариантах данной задачи. А именно, изначально в исходной задаче было полностью убрано одно из трех условий ортогональности – т.е. осуществлялся поиск такой тройки ортогональных латинских квадратов $A, B, C$, что $A$ ортогонален $B$, $A$ ортогонален $C$, а ортогональность между $B$ и $C$ не требовалась. КНФ, кодирующая данную задачу, состоит из 23000 переменных и 1091631 дизъюнктов, файл с КНФ занимает 17 мегабайт. В каждом квадрате была зафиксирована первая строка (значение «0 1 2 3 4 5 6 7 8 9»). Декомпозиция проводилась по второй строке первого квадрата – перебирались все возможные варианты ее заполнения (всего 1334961 вариантов, каждому варианту соответствует подзадача). Эксперимент по решению данной задачи в SAT@home был запущен 24 июня 2013 года и продолжался примерно два месяца. В результате нашлась такая тройка $A, B, C$, что $A$ ортогонален $B$, $A$ ортогонален $C$, а квадраты $B$ и $C$ ортогональны в 71 ячейке из 100. После этого был запущен поиск тройки квадратов, в которой на $B$ и $C$ было наложено условие «ортогональны минимум в 72 ячейках из 100». КНФ, кодирующая данную задачу, состоит из 36685 переменных и 1625560 дизъюнктов, файл с КНФ занимает 23 мегабайта. Эта задача решается в проекте в настоящий момент.

В дальнейшем мы планируем разработать и реализовать новые кодировки для приведенных в статье задач, а также ускорить работу используемого SAT-решателя на задачах поиска систем ортогональных латинских квадратов.

## Список литературы


1. Colbourn C.J., Dinitz J.H. Handbook of Combinatorial Designs. Second Edition. Chapman&Hall, 2006. 984 p.

2. Biere A., Heule V., van Maaren H., Walsh T. (eds.) Handbook of Satisfiability. IOS Press, 2009.

3. Zhang H. Combinatorial Designs by SAT Solvers. In: Biere et al. [2], pp. 533-568.

4. C. Gomes and D. Shmoys. Completing quasigroups or latin squares: A structured graph coloring problem. In Proceedings of the Computational Symposium on Graph Coloring and Generalizations, pp. 22-39, 2002.

5. Brown et al. Completion of the Spectrum of Orthogonal Diagonal Latin Squares. Lecture notes in pure and applied mathematics. 1992. Vol. 139. pp. 43–49.

6. Заикин О.С., Посыпкин М.А., Семёнов А.А., Храпов Н.П. Опыт организации добровольных вычислений на примере проектов OPTIMA@home и SAT@home // Вестник ННГУ. № 5(2). 2012. С. 338-346.

7. Заикин О.С., Семенов А.А., Посыпкин М.А. Процедуры построения декомпозиционных множеств для распределенного решения SAT-задач в проекте добровольных вычислений SAT@home // Управление большими системами. Выпуск 43. М.: ИПУ РАН, 2013. С. 138-156.

8. SAT@home: проект добровольных вычислений для решения крупномасштабных SAT-задад. URL: http://sat.isa.ru/pdsat/

9. Anderson, D.P. BOINC: A System for Public-Resource Computing and Storage // In: Buyya, R. (ed.) GRID. pp. 4-10. IEEE Computer Society, 2004.

10. Rainbow tables for A5/1, http://opensource.srlabs.de/projects/a51-decrypt

11. Een, N., Sorensson, N. An Extensible SAT-solver. In: Giunchiglia, E., Tacchella, A. (eds.) SAT. LNCS, vol. 2919, pp. 502-518. Springer (2003).