МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

МАТЕРИАЛЫ

XI-й Международной молодёжной научной конференции

«МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННЫХ, ТЕХНИЧЕСКИХ И ЭКОНОМИЧЕСКИХ СИСТЕМ»

Томск, 24–27 мая 2024 г.

Под общей редакцией
кандидата технических наук И.С. Шмырина

Томск
Издательство Томского государственного университета
2024

ББК 22.17– 22.19
УДК 519.2, 519.7, 519.8
Т78

ЧЛЕНЫ КОЛЛЕГИИ, РУКОВОДИТЕЛИ НАУЧНЫХ РЕДАКЦИЙ
ПО НАПРАВЛЕНИЯМ:

д-р физ.-мат. наук, проф. **А.А. Глазунов** – научная редакция «Механика, математика»; д-р физ.-мат. наук, проф. **Э.Р. Шрагер** – научная редакция «Механика, математика»; д-р техн. наук, проф. **С.П. Сущенко** – научная редакция «Информатика и кибернетика»; д-р физ.-мат. наук, проф. **В.Г. Багров** – научная редакция «Физика»; д-р физ.-мат. наук, проф. **А.И. Потекаев** – научная редакция «Физика»; д-р биол. наук, проф. **С.П. Кулижский** – научная редакция «Биология»; д-р геол.-минер. наук, проф. **В.П. Парначев** – научная редакция «Науки о Земле, химия»; канд. хим. наук, доц. **Ю.Г. Слижов** – научная редакция «Науки о Земле, химия»; д-р филол. наук, проф. **Т.А. Демешкина** – научная редакция «История, филология»; д-р ист. наук, проф. **В.П. Зиновьев** – научная редакция «История, филология»; д-р экон. наук, проф. **В.А. Уткин** – научная редакция «Юридические и экономические науки»; д-р ист. наук, проф. **Э.И. Черняк** – научная редакция «Философия, социология, психология, педагогика, искусствознание»; д-р психол. наук, проф. **Э.В. Галажинский** – научная редакция «Философия, социология, психология, педагогика, искусствознание»

НАУЧНАЯ РЕДАКЦИЯ ТОМА:

д-р техн. наук, проф. **А.В. Замятин**, д-р техн. наук, проф. **С.П. Сущенко**, д-р физ.-мат. наук, проф. **Л.А. Нежельская**, д-р физ.-мат. наук, доц. **А.Н. Моисеев**, д-р физ.-мат. наук, проф. **С.П. Моисеева**, канд. техн. наук, доц. **Н.Л. Ерёмина**, канд. техн. наук, доц. **С.А. Останин**, канд. техн. наук **И.С. Шмырин**.

Т78 Труды Томского государственного университета. – Т. 309. Серия физико-математическая: Математическое и программное обеспечение информационных, технических и экономических систем : материалы XI-й Международной молодѐжной научной конференции. Томск, 24–27 мая 2024 г. / под общ. ред. И.С. Шмырина. – Томск : Издательство Томского государственного университета, 2024. – 324 с.

ISBN 978-5-907890-15-2

Сборник содержит материалы XI-й Международной молодѐжной научной конференции «Математическое и программное обеспечение информационных, технических и экономических систем», проводившейся 24–27 мая 2024 г. на базе Института прикладной математики и компьютерных наук Томского государственного университета. Материалы сгруппированы в соответствии с работавшими на конференции секциями.

Для научных работников, преподавателей, аспирантов, магистрантов и студентов.

УДК 539.3.004
ББК 22.17-22.19

ISBN 978-5-907890-15-2
© Томский государственный университет, 2024

Из полученных в ходе эксперимента результатов видно, что скорость внедрения при параллельной обработке по кадрам выше скорости внедрения при параллельной внутрикадровой обработке. Но при малом количестве потоков, для ряда процессоров, скорость внутрикадрового распараллеливания может быть выше покадрового. Таким образом, если проектируемая система имеет возможность покадровой параллельной обработки, то целесообразно использовать еѐ. Если количество возможных потоков покадрового распараллеливания меньше числа независимых потоков процессора, то имеет смысл осуществлять внутрикадровое распараллеливание с целью максимально задействовать вычислительные ресурсы процессора.

Заключение

В рамках эксперимента получены оценки скорости внедрения водяного знака E_BLIND/D_LC с использованием различных подходов к распараллеливанию. Полученные результаты позволяют сформулировать следующие рекомендации по параллельному встраиванию водяных знаков.

При параллельной обработке данных внутри фрейма наиболее эффективным оказывается его разбиение по группам горизонтальных строк пикселей. При параллельной обработке, организованной по отдельным фреймам, скорость внедрения оказывается быстрее по сравнению с параллельной обработкой, организованной по данным внутри каждого из фреймов. Таким образом, если проектируемая система имеет возможность покадровой параллельной обработки, то целесообразно использовать еѐ. Если количество возможных потоков покадрового распараллеливания меньше числа независимых потоков процессора, то имеет смысл осуществлять внутрикадровое распараллеливание с целью максимально задействовать вычислительные ресурсы процессора.

ЛИТЕРАТУРА

1. *Wagner N.R.* Fingerprinting // 1983 IEEE Symposium on Security and Privacy. – IEEE, 1983. – P. 18.
2. *Орешкина Е.И., Фаворская М.Н.* Классификация методов нанесения цифровых водяных знаков // Актуальные проблемы авиации и космонавтики. – 2015. – Т. 1. – №. 11. – С. 414–415.
3. *Гергель В.П.* Теория и практика параллельных вычислений. – 2007.
4. *Cox I. et al.* Digital watermarking and steganography. – Morgan kaufmann, 2007.
5. Watermarks embedding performance evaluation and optimization // GitHub URL: https://github.com/anjinviktor/watermarking_performance (дата обращения: 05.05.2024).
6. *Asikuzzaman M., Pickering M.R.* An overview of digital video watermarking // IEEE Transactions on Circuits and Systems for Video Technology. – 2017. – V. 28. – №. 9. – P. 2131–2153.
7. *Kowarschik M., Weiß C.* An overview of cache optimization techniques and cache-aware numerical algorithms // Algorithms for memory hierarchies: advanced lectures. – 2003. – P. 213–232.

МИНИМИЗАЦИЯ КОРНЯ ЛОГИЧЕСКОГО УРАВНЕНИЯ КНФ = 1

Бирюков М.О., Андреева В.В.

Томский государственный университет

iilll@duck.com, avv.21@mail.ru

Введение

Задача выполнимости булевых формул (SAT-задача) – фундаментальная задача в области информатики и теории вычислительной сложности. Она является первой, для которой удалось доказать ее принадлежность к классу NP-полных задач в начале 1970-х гг. (при формулировке относительно КНФ). Она имеет широкое применение в ряде областей, включая искусственный интеллект, верификацию программного обеспечения, криптографию, проектирование интегральных схем [1]. Активные исследования в еѐ направлении ведутся по сей день, ежегодно проводится международное соревнование SAT-решателей (программных реализаций алгоритмов решения SAT-задачи) [2].

Большой класс существующих полных алгоритмов решения SAT-задачи основывается на методе ветвей и границ (поиске с возвратом), при котором поиск корня логиче-

ского уравнения КНФ = 1 осуществляется через последовательное "разветвление" пространства поиска путем фиксирования значения переменных, входящих в КНФ, и рекурсивного рассмотрения каждой из этих ветвей [1].

Одним из первых и наиболее известных алгоритмов решения SAT-задачи является DPLL-алгоритм, разработанный в 1961 г., в основе которого лежит метод ветвей и границ вместе с двумя правилами логического вывода: правила единичного дизъюнкта и правила чистой литеры.

Независимо от DPLL, советским и белорусским кибернетиком А.Д. Закревским в 1964 г. был разработан во многом схожий алгоритм решения SAT-задачи [3], который будем называть алгоритмом Закревского.

Одним из наиболее эффективных алгоритмов решения SAT-задачи на сегодняшний день считается алгоритм CDCL, представляющий собой улучшение DPLL.

В некоторых практических задачах возникает необходимость в решении смежной задачи минимизации корня логического уравнения КНФ = 1. Так, например, в [4] предложен метод построения тестовых наборов для неисправности схемы, который сводится к поиску допустимых интервалов некоторой КНФ и формированию проверяющего теста на их основе. При этом, чем меньше ранг найденных интервалов, тем больше возможностей для сокращения длины проверяющего теста, что является предпочтительным.

В данной работе разработан стохастический метод минимизации корня логического уравнения вида КНФ = 1 на основе алгоритма CDCL.

1. Постановка задачи и существующие методы ее решения

Формально задача минимизации корня логического уравнения вида КНФ = 1 представляет собой задачу комбинаторной оптимизации, заключающуюся в поиске для заданной КНФ допустимого интервала (корня) как можно меньшего ранга. Данная задача во многом схожа с задачей выполнимости булевых формул, ввиду чего для нее по большей части применимы результаты из этой области.

Из различий рассматриваемой задачи с задачей выполнимости булевых формул важно отметить, что пространство поиска представляет собой не n-мерное булево пространство B^n, где n – количество переменных рассматриваемой КНФ, а множество всевозможных интервалов в этом булевом пространстве, которое обозначим V_n.

Из существующих методов минимизации корня логического уравнения КНФ = 1 обратим внимание на метод, предложенный в [5] и представляющий собой модификацию алгоритма Закревского. Основная модификация состоит в специальной эвристике ветвления, направленной на поиск корня как можно меньшего ранга, и основная идея которой заключается в том, чтобы в троичной матрице, соответствующей рассматриваемой КНФ, в первую очередь покрывать строки минимального ранга, т.к. они в наибольшей степени влияют на вид корня (допустимого интервала), если он существует.

Эффективность этого метода на случайно сгенерированных КНФ в сравнении с исходным алгоритмом Закревского была показана экспериментально [5], однако данный метод все же обладает несколькими недостатками.

В первую очередь, ввиду природы механизма поиска DPLL-подобных алгоритмов (включая алгоритм Закревского), часть пространства поиска V_n рассматриваемой оптимизационной задачи неизбежно упускается из рассмотрения. Это происходит ввиду того, что формально поиск корня через метод ветвей и границ в DPLL-подобных алгоритмах сводится к обходу "дерева поиска", которое представляет собой направленный ациклический граф, состоящий из $n+1$ уровней (n – количество переменных в КНФ), где k-й уровень содержит 2^k вершин, соответствующих различным интервалам ранга k в B^n. Так, суммарное количество вершин этого дерева (точек пространства поиска), кото-

рые возможно рассмотреть в ходе поиска, меньше, чем мощность пространства поиска V_n рассматриваемой оптимизационной задачи (1):

$$\sum_{k=0}^{n} 2^k < \sum_{k=0}^{n} C_n^k \cdot 2^k \ , \ n \in N \ . \qquad (1)$$

На рис. 1 представлен пример одного из возможных полных деревьев поиска для двумерного булева пространства; на рис. 2 изображен направленный ациклический граф ("граф поиска" для B^2), объединяющий в себя всевозможные деревья поиска для двумерного булева пространства B^2. Серым помечены вершины (точки пространства поиска), которые упускаются из рассмотрения при обходе дерева поиска, как на рис. 1.

Также данный алгоритм, как и все DPLL-подобные алгоритмы, не применяющие периодический перезапуск поиска, подвержен "застреванию" в неудачно выбранных ветвях (областях), являющихся значительной долей огромного пространства поиска, но при этом не содержащих корней [1,6].

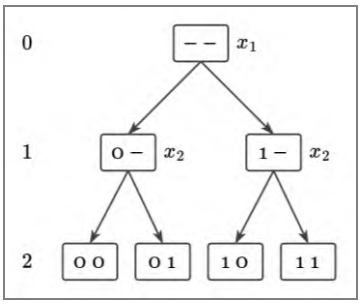

Рис. 1. Одно из возможных деревьев поиска для B^2

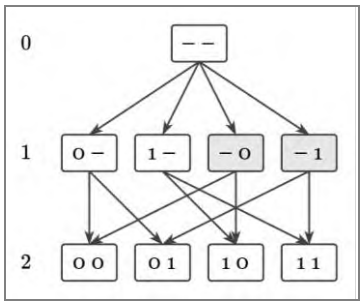

Рис. 2. Граф поиска для B^2

2. Стохастический метод минимизации корня логического уравнения КНФ = 1

2.1. Описание метода

С учетом недостатков предыдущего метода в данной работе поиск корня логического уравнения вида КНФ = 1 как можно меньшего ранга (минимизацию корня) предлагается осуществлять через модифицированный алгоритм CDCL с периодическим перезапуском поиска и со стохастической эвристикой ветвления. Модификация CDCL состоит в нескольких небольших изменениях, адаптирующих этот алгоритм к рассматриваемой оптимизационной задаче:

1. Исходная КНФ проверяется на выполнимость в каждом узле дерева поиска, т.е. для каждого рассматриваемого промежуточного решения (набора присвоений значений переменным, интервала).

2. Поиск продолжается до тех пор, пока не выполнится заданное условие останова. При нахождении корня он добавляется в список найденных корней, и после хронологического отката на предыдущий не до конца рассмотренный узел принятия решения поиск продолжается дальше. При этом узлы дерева поиска, лежащие на уровне больше или равным минимальному рангу из найденных корней, отмечаются как не содержащие решений.

3. Для найденных интервалов (корней) предпринимается попытка их расширения за счет удаления некоторых внешних компонент.

4. Подобно алгоритму Закревского, рассматриваемая в ходе поиска КНФ представляется в виде троичной матрицы. Выбор переменной ветвления и еѐ значения в описанной ниже стохастической эвристике ветвления опирается только на анализ этой троичной матрицы.

5. Используется стохастическая эвристика ветвления, направленная на поиск корня как можно меньшего ранга.

2.2. Стохастическая эвристика ветвления

Основой предлагаемого метода является стохастическая эвристика ветвления, при которой выбор переменной ветвления и еѐ значения определяется реализацией дискретной случайной величины H со множеством значений $X \times B$ (X– множество переменных КНФ, B – булево множество), распределение вероятностей которой формируется на основе анализа троичной матрицы, соответствующей текущей упрощенной КНФ. Такая эвристика позволяет отразить в себе весь спектр "весов" или "качеств" возможных выборов, а не какой-то один выбор, соответствующий максимальному "весу", как при детерминированной эвристике ветвления рассмотренного выше метода, основанного на алгоритме Закревского.

В совокупности же с периодическим перезапуском поиска с начала, появляется возможность рассмотрения нескольких близких по "предпочтительности" деревьев поиска, за счет чего более тщательно исследуется пространство поиска V_n.

Пусть на данном этапе выбора текущая упрощенная КНФ содержит m дизъюнктов, а соответствующая ей троичная матрица обозначена как U. Пусть также исходная КНФ зависит от множества переменных X мощности n.

Итоговое распределение вероятностей случайной величины H строится в несколько этапов. В первую очередь, для каждой пары переменной и еѐ значения $x_k = a$ (выбора) из множества $X \times B$ вычисляется соответствующий ей неотрицательный "вес" с помощью функции оценки $\sigma : X \times B \to [0, +\infty)$, представленной в (2):

$$\sigma(x_k, a) = \sum_{s=1}^{m} I\{u_{s,k} = a\}\,\omega(u_s), \tag{2}$$

где I – индикаторная функция, u_s – s-я строка матрицы U, $u_{s,k}$ – k-я компонента s-й строки матрицы U, $\omega(u_s)$ – "вес" или "значимость" s-ой строки матрицы U. Так, для выбора переменной ветвления и ее значения $x_k = a$ величина $\sigma(x_k, a)$ представляет собой сумму весов обращающихся в истину дизъюнктов (строк) при данном назначении переменной, где вес дизъюнкта определяется рангом соответствующей ему строки: чем меньше ранг, тем больше вес.

В качестве весовой функции ω предлагается взять (3):

$$\omega(c) = 2^{-\mathrm{rang}(c)}. \tag{3}$$

В (3) вес ω(c) дизъюнкта (строки) c равен отношению мощности соответствующего ему интервала из множества нулевых значений исходной КНФ к мощности всего булева пространства B^n.

Для всех $x \in X$ из пар значений $\sigma(x, 0)$ и $\sigma(x, 1)$ предлагается рассматривать только максимальное из них – большее из этих значений берется неизменным, а оставшееся принимается равным нулю. В результате такого "прореживания" в ноль обращаются n из исходных $2n$ значений весов. Наконец, из результирующего набора весов $\{\sigma(x, a) : (x, a) \in X \times B\}$ формируется итоговое распределение вероятностей через "нормирующее" преобразование f с параметром $\gamma > 0$ (4):

$$f_\gamma(\sigma(x, a)) = \frac{\sigma(x, a)^\gamma}{\sum\limits_{(x,a) \in X \times B} \sigma(x, a)^\gamma}. \tag{4}$$

Важным свойством преобразования (4) является то, что $f_\gamma(0) = 0$ при $\gamma > 0$, т.е., выбор переменной ветвления и ее значения, которому при функции оценки вида (2) соответствует нулевой вес, будет иметь нулевую вероятность быть "выбранным" данной стохастической эвристикой ветвления. Это необходимо для того, чтобы эвристика выбирала только те переменные для ветвления, которые не входят в текущее промежуточное решение, и которым в упрощенной КНФ (троичной матрице) соответствуют

столбцы, содержащие хотя бы одну определенную компоненту. Так, вектор (набор) итогового распределения вероятностей длины $2n$ содержит как минимум $n+r$ нулевых компонент и выбор сводится к $n-r$ вариантам, где r – число назначений переменным в текущем промежуточном решении.

Параметр γ преобразования (4) позволяет варьировать "контраст" результирующего распределения вероятностей:

– при $\gamma \in (0,1)$ результирующее распределение вероятностей "приближается" к равновероятному, ненулевые компоненты которого равны $(n-r)^{-1}$;
– при $\gamma = 1$ преобразование (4) вырождается в нормирование вектора весов по норме l_1, при котором компоненты результирующего вектора распределения вероятностей пропорциональны соответствующим им весам;
– при $\gamma > 1$, подобно многопеременной логистической функции (softmax), в большей степени выражается максимальная компонента распределения.

На рис. 3, 4 приведены примеры распределений вероятностей, полученных через описанный выше алгоритм при разных значениях параметра γ для КНФ (троичной матрицы) из табл. 6 работы [5] (нулевые компоненты не изображены).

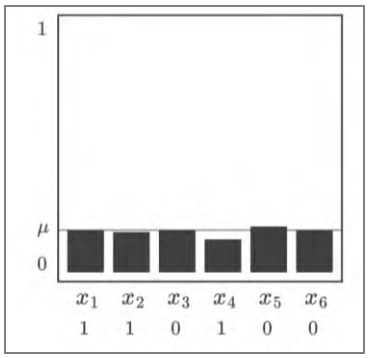

Рис. 3. Распределение вероятностей при γ = 1/4

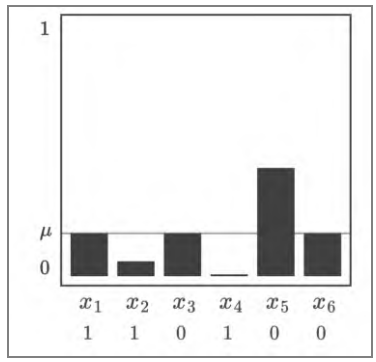

Рис. 4. Распределение вероятностей при γ = 4

3. Экспериментальные результаты

Разработанный алгоритм был реализован на языке C++ и испытан на различных бенчмарках (КНФ) [7], включающих в себя равномерно случайные КНФ (uf*, uuf*), состоящие из трех равновероятно взятых литер, а также "индустриальные" бенчмарки, соответствующие реальным задачам: задаче индуктивного вывода (ii*) и задаче проверки наличия неисправности в логической схеме (ssa*) (табл. 1–3).

Таблица 1

Результаты работы разработанного метода на индустриальных бенчмарках

b	*n*	*m*	*r*	*c*	*s*	*rs*
ii8a2	180	800	143	9842	514 445	157
ii8a3	264	1552	191	12 352	165 018	62
ii8a4	396	2798	283	14 794	30 450	22
ii8b1	336	2068	191	815	101 844	44
ii8b4	1068	8214	658	2289	1155	2
ssa.7552.038	1501	3575	1448	1514	80 927	31
ssa.7552.158	1363	3034	1327	1718	117 013	48
ssa.7552.159	1363	3032	1327	1988	124 188	62
ssa.7552.160	1391	3126	1359	3776	86 833	45

Примечания:
n – количество переменных КНФ;
m – количество дизъюнктов КНФ;
r – минимальный ранг из найденных в течение 3-х минут корней;
c – количество встреченных конфликтов;

s – количество "пропусков" областей ниже минимального ранга; *rs* – количество раз был произведен перезапуск поиска с начала; *b* – название бенчмарка (КНФ).

Таблица 2

Результаты работы предыдущего метода на индустриальных бенчмарках

b	*n*	*m*	*r*	*c*	*i*	*t*
ii8a2	180	800	159	1 443 750	12 588 158	723 854
ii8a3	264	1552	–	568 375	9 516 214	900 002
ii8a4	396	2798	349	20	681	328
ii8b1	336	2068	191	7448	205 704	26 772
ii8b4	1068	8214	–	30 071	1 729 557	900 010
ssa.7552.038	1501	3575	–	44 733	1 471 992	900 008
ssa.7552.158	1363	3034	–	57 515	1 853 963	900 002
ssa.7552.159	1363	3032	1327	38	2753	1781
ssa.7552.160	1391	3126	–	64 681	2 339 482	900 002

Примечания:
n – количество переменных КНФ;
m – количество дизъюнктов КНФ;
r – ранг найденного корня (прочерк, если не найден за 15 минут);
c – количество встреченных конфликтов;
i – количество имплицированных присвоений;
t – время работы алгоритма в миллисекундах;
b – название бенчмарка (КНФ).

Таблица 3

Результаты работы двух методов на равномерно случайных и невыполнимых КНФ

b	*n*	*m*	*r*	*r**	*t*	*t**
uf.125.538 (100)	125	538	116,74	116,09	1 123,93	3 494,07
uf.150.645 (64)	150	645	140,53	139,61	4 079,96	11 121,98
uuf.150.645.1	150	645	–	–	900 008 †	547
uuf.200.860.1	200	860	–	–	900 002 †	1 136
ssa.0432.003	435	1027	–	–	5 996	189
ssa.2670.141	986	2315	–	–	900 002 †	3498

Примечания:
n – количество переменных КНФ;
m – количество дизъюнктов КНФ;
r – средний ранг первого найденного корня предыдущим методом;
*r** – средний ранг первого найденного корня новым методом;
t – среднее время работы предыдущего метода в миллисекундах;
*t** – среднее время работы нового метода в миллисекундах;
b – название бенчмарка (КНФ).
Символ "†" означает, что алгоритм не пришел к выводу о невыполнимости КНФ по истечении 15-ти минут.

В приведенных в табл. 2 результатах работы предыдущего метода наблюдается эффект "застревания" DPLL-подобных алгоритмов, не применяющих периодический перезапуск поиска, в неудачно выбранных "плохих" областях пространства поиска: для некоторых КНФ удается найти корень за относительно короткое время, а на других алгоритм уходит в длительный и безрезультатный поиск.

На выполнимых равномерно случайных 3-КНФ оба алгоритма показывают близкие к равным результаты.

В табл. 3 также представлено время работы двух алгоритмов на невыполнимых КНФ, из которого можно видеть, что разработанный метод, во-первых, остается полным алгоритмом решения SAT-задачи, и, во-вторых, благодаря анализу конфликтов позволяет прийти к выводу о невыполнимости формулы быстрее, чем предыдущий метод.

Наконец, обращая внимание на результаты работы двух методов на некоторых индустриальных бенчмарках в табл. 1 и 2, можно видеть, что, по крайней мере, на рассмотренных КНФ разработанный в данной работе стохастический метод минимизации корня логического уравнения вида КНФ = 1 позволяет находить корни меньшего ранга

и за меньшее время, чем рассмотренный существующий метод, основанный на алгоритме Закревского.

Заключение

В данной работе была рассмотрена задача минимизации корня логического уравнения вида КНФ = 1, а также смежная с ней задача выполнимости булевых формул в формулировке КНФ. Также был рассмотрен один из существующих методов минимизации корня уравнения КНФ = 1, основанный на алгоритме Закревского. Были выявлены несколько его недостатков.

На основе этого анализа был разработан стохастический метод минимизации корня логического уравнения вида КНФ = 1, представляющий собой модифицированный алгоритм CDCL со стохастической эвристикой ветвления.

Разработанный метод был реализован на языке C++, его эффективность в сравнении с предыдущим методом показана экспериментально.

ЛИТЕРАТУРА

1. *Biere A., Heule M., Van Maaren H., Walsh T.* Handbook of Satisfiability // Frontiers in Artificial Intelligence and Applications. – IOS Press, 2021. – P. 336. – ISBN 978-1-64368-160-3.
2. The International SAT Competition Web Page // URL: https://satcompetition.github.io; (дата обращения 25.05.2024).
3. *Закревский А.Д.* Логические уравнения. – М.: Едиториал УРСС, 2003. – 96 с.
4. *Тычинский В.З., Андреева В.В.* Получение тестовых пар для робастно тестируемых неисправностей задержек путей с использованием SAT- решателей // Математическое и программное обеспечение информационных, технических и экономических систем : материалы Международной научной конференции, Томск, 28–30 мая 2020 г. – Томск: Изд-во Том. ун-та, 2020. – С. 194–200.
5. *Андреева В.В., Тарновская Т.П.* Сокращение ранга конъюнкции, представляющей корень логического уравнения // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. – 2015. – №4 (33). – С. 62–68.
6. *Gomes C.P., Selman B., Kautz H.* Boosting combinatorial search through randomization // AAAI/IAAI 98.1998 (1998): P. 431–437.
7. SATLIB – Benchmark problems // URL: https://www.cs.ubc.ca/~hoos/SATLIB/benchm.html; (дата обращения 25.05.2024).

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ КАТАЛОГИЗАЦИИ АУДИОКНИГ

Гайсин Р.Е., Пахомова Е.Г.

Томский государственный университет

gaisin.roman@mail.ru, PakhomovaEG@yandex.ru

Введение

В последние годы аудиокниги стали весьма популярными благодаря своей практичности и доступности [1]. Они позволяют наслаждаться литературой в ситуациях, когда чтение печатных или электронных книг невозможно, например, при вождении автомобиля, занятиях спортом или выполнении домашних дел. С ростом количества аудиокниг появляется потребность в разработке эффективных средств для их каталогизации и управления.

Целью данной работы является разработка эффективного и удобного инструмента, который позволит пользователям легко управлять своей коллекцией аудиокниг. Приложение будет обеспечивать удобную организацию, хранение, поиск, фильтрацию и легкий доступ к метаданным аудиокниг. Помимо этого, оно должно обладать удобным и интуитивно понятным интерфейсом.

1. Проектирование приложения

Архитектура приложения. Для приложения была выбрана архитектура MVC. Причиной стала возможность четкого разделения ответственности между данными,