

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**РАЗРАБОТКА ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ РЕШЕНИЯ ЗАДАЧИ
БУЛЕВОЙ ВЫПОЛНИМОСТИ С ИСПОЛЬЗОВАНИЕМ ПОИСКА
ВЕРОЯТНОСТНЫХ ЛАЗЕЕК**

Автор: Джиблави Ибрагим Билалович _____

Направление подготовки: 01.03.02 Прикладная
математика и информатика

Квалификация: Бакалавр

Руководитель ВКР: Чивилихин Д.С., к.т.н. _____

Санкт-Петербург, 2022 г.

Обучающийся Джиблavi Ибрагим Билалович
Группа М3439 Факультет ИТиП

Направленность (профиль), специализация
Математические модели и алгоритмы в разработке программного обеспечения

ВКР принята « ____ » _____ 20 ____ г.

Оригинальность ВКР ____ %

ВКР выполнена с оценкой

Дата защиты «15» июня 2022 г.

Секретарь ГЭК Павлова О.Н.

Листов хранения

Демонстрационных материалов/Чертежей хранения

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

УТВЕРЖДАЮ

Руководитель ОП
проф., д.т.н. Парфенов В.Г. _____
« ____ » _____ 20__ г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ

Обучающийся Джиглави Ибрагим Билалович

Группа М3439 **Факультет** ИТиП

Квалификация: Бакалавр

Направление подготовки: 01.03.02 Прикладная математика и информатика

Направленность (профиль) образовательной программы: Математические модели и алгоритмы в разработке программного обеспечения

Тема ВКР: Разработка параллельных алгоритмов решения задачи булевой выполнимости с использованием поиска вероятностных лазеек

Руководитель Чивилихин Д.С., к.т.н., ординарный доцент

2 Срок сдачи студентом законченной работы до: «15» мая 2022 г.

3 Техническое задание и исходные данные к работе

Ключевой задачей является разработка эффективного алгоритма решения задач булевой выполнимости. Алгоритм по заданной формуле должен возвращать информацию о её выполнимости. Подразумевается параллельный алгоритм, использующий поиск вероятностных лазеек для сведения задачи к набору более простых подзадач и последующему решению этих подзадач с применением существующих решателей.

4 Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов)

В данной работе разработаны алгоритмы поиска вероятностных лазеек на основе эволюционных алгоритмов. Разработаны различные подходы к решению на основе методов поиска вероятностных лазеек, исследована их эффективность. Проведено сравнение эффективности с существующими конкурентоспособными алгоритмами.

5 Перечень графического материала (с указанием обязательного материала)

Графические материалы и чертежи работой не предусмотрены

6 Исходные материалы и пособия

а) TBD.

7 Дата выдачи задания «31» января 2022 г.

Руководитель ВКР _____

Задание принял к исполнению _____ «31» января 2022 г.

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Обучающийся: Джиблavi Ибрагим Билалович

Наименование темы ВКР: Разработка параллельных алгоритмов решения задачи булевой выполнимости с использованием поиска вероятностных лазеек

Наименование организации, в которой выполнена ВКР: Университет ИТМО

ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: TBD

2 Задачи, решаемые в ВКР:

- а) разработка эффективных алгоритмов поиска вероятностных лазеек.
- б) разработка методов решения задачи булевой выполнимости с применением вероятностных лазеек.
- в) исследование эффективности полученных алгоритмов.

3 Число источников, использованных при составлении обзора: 0

4 Полное число источников, использованных в работе: 11

5 В том числе источников по годам:

Отечественных			Иностраннх		
Последние 5 лет	От 5 до 10 лет	Более 10 лет	Последние 5 лет	От 5 до 10 лет	Более 10 лет
2	3	5	0	1	0

6 Использование информационных ресурсов Internet: да, число ресурсов: 2

7 Использование современных пакетов компьютерных программ и технологий:

Пакеты компьютерных программ и технологий	Раздел работы
Что	Раздел работы

8 Краткая характеристика полученных результатов

TBD

9 Гранты, полученные при выполнении работы

TBD

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы

TBD

Обучающийся Джиблavi И.Б. _____

Руководитель ВКР Чивилихин Д.С. _____

« ____ » _____ 20 ____ г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1. Обзор.....	7
1.1. Термины и понятия.....	7
1.2. Последовательные решатели	8
1.3. Параллельные решатели.....	8
Выводы по главе 1	8
2. Теоретическое исследование и архитектура.....	9
2.1. Поиск вероятностных лазеек	9
2.1.1. Схема алгоритма	9
2.1.2. Эффективный вывод последствий	10
2.2. Решение набора подстановок.....	13
2.3. Использование вероятностных лазеек	15
2.4. Описание реализации	15
Выводы по главе 2	15
3. Практическое исследование	16
Выводы по главе 3	16
ЗАКЛЮЧЕНИЕ	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	18

ВВЕДЕНИЕ

Задача булевой выполнимости является крайне важной NP-полной [4] алгоритмической задачей, так как к ней сводится большое число задач. Среди них:

- Проверка моделей, использующийся для формальной верификации параллельных систем с конечным числом состояний [1].
- Восстановление секретного ключа в алгоритме RSA [9].
- Разложение булевых матриц [2], часто применяемое в рекомендательных системах.
- 0-1 задача о рюкзаке (широко известная задача класса NP).
- Многие задачи планирования, например, one shop scheduling [10], сводятся к SAT.

Для задачи булевой выполнимости, как и для любой NP-полной задачи пока не известно алгоритма, решающего её за полиномиальное от размера входа время. Однако, спектр применения этой задачи достаточно широк, чтобы возникала практическая польза от разработки эффективных решателей.

Алгоритмы решения задачи булевой выполнимости глобально делятся на две категории. Первая — последовательные решатели, то есть работающие в одном потоке исполнения. Существующие эффективные и широко распространенные подходы к построению таких решателей более подробно описаны в Главе 1.2. Вторая — параллельные решатели, нацеленные на максимальную утилизацию доступных ресурсов многоядерных и многопроцессорных машин для ускорения процесса решения. Более подробно они описаны в Главе 1.3.

Целью данной работы является построение эффективного параллельного алгоритма решения задачи булевой выполнимости на основе поиска вероятностных лазеек. В рамках исследований в TBD-REF данное семейство подходов показало положительные результаты, однако остается достаточно большое пространство как для построения более конкурентноспособной реализации, так и для расширения семейства подходов.

В рамках данной работы поставлены и выполнены следующие задачи:

- Разработка эффективного алгоритма поиска вероятностных лазеек на основе подходов, изложенных в TBD-REF.
- Адаптация набора существующих решателей для использования в рамках разрабатываемых алгоритмов.

- Разработка альтернативных предложенным в TBD-REF схем решения с использованием вероятностных лазеек.
- Исследование эффективности реализованных схем и сравнение с существующими параллельными решателями.

В Главе 1.1 содержатся необходимые понятия и определения. В Главе 1.2 поверхностно рассмотрены существующие эффективные и широко распространенные подходы к реализации последовательных решателей. В Главе 1.3 более подробно описаны популярные и успешные стратегии работы параллельных решателей.

В Главе 2.1 рассмотрен вопрос поиска вероятностных лазеек. В разделе 2.1.1 построены эффективные эволюционные алгоритмы для решения этой задачи. В разделе 2.1.2 описаны изменения, внесенные в исходный код Minisat, позволяющие существенно ускорить процесс поиска решения. В Главе 2.2 описан построенный параллельный сервис для решения задач булевой выполнимости. Описан процесс адаптации и унификации существующих последовательных решателей, способы обмена знаниями между решателями. В Главе 2.3 приведено описание и описана реализация как упоминаемых в TBD-REF, так и новых стратегий использования вероятностных лазеек в решателе.

ГЛАВА 1. ОБЗОР

1.1. Термины и понятия

Определение 1. Булевой формулой¹ называется формула логики высказываний, содержащая логические переменные и пропозициональные связки \wedge, \vee, \neg . Множество логических переменных обозначается V .

Определение 2. Булевой функцией называется отображение $E: \mathcal{B}^n \rightarrow \mathcal{B}$, где $\mathcal{B} = \{0, 1\}$, а n — число различных переменных. Булева формула F задает булеву функцию E_F . Далее разница между этими понятиями для нас несущественна, поэтому всегда будет упоминаться булева функция E .

Определение 3. Задача булевой выполнимости (SAT) — проверить, существует ли $x \in \mathcal{B}$ такой, что выполняется $E(x) = 1$.

Теорема 4. (Кук, Левин) [4] Задача булевой выполнимости принадлежит классу NP -полных задач.

Определение 5. Решателем, алгоритмом A называется алгоритм, принимающий на вход описание булевой формулы C и выдающий результат проверки булевой выполнимости соответствующей булевой функции — $A(C)$. Возможны следующие результаты:

- 0, функция невыполнима.
- 1, функция выполнима. В этом случае также может быть возвращено удовлетворяющее функции назначение переменных R : $E(V \mid R) = 1$.
- ?, решатель не смог решить задачу либо вследствие его неполноты, либо из-за нехватки ресурсов, таких как время или память.

Обозначим за $S(A)$ множество булевых функций, разрешимых алгоритмом A за полиномиальное время.

Определение 6. Означиванием, подстановкой набора переменных $B \subseteq V$ называется отображение $\hat{b}: B \rightarrow \mathcal{B}$. Означивание можно применить к булевой функции E , результатом будет другая булева функция $E[B \mid \hat{b}]: \mathcal{B}^{|V|-|B|} \rightarrow \mathcal{B}$, возможно тождественная. Она получается из исходной путем частичной подстановки переменных, назначенных функцией \hat{b} . Множество всех означиваний множества переменных B обозначается \hat{B} .

Определение 7. Лазейкой называется множество $B \subseteq V$ такое, что для любого из $2^{|B|}$ означиваний \hat{b} переменных из B выполняется $E[B \mid \hat{b}] \equiv 0$.

¹https://ru.wikipedia.org/wiki/Булева_формула

Ясно, что если для функции E существует лазейка, то она невыполнима, то есть $E \equiv 0$.

Определение 8. ρ -Лазейкой называется множество $B \subseteq V$ такое, что выполняется

$$\left| \left\{ \hat{b} \mid \hat{b} \in \hat{B}, E[B \mid \hat{b}] \equiv 0 \right\} \right| \geq \rho \cdot 2^{|B|}.$$

Определение 9. (ε, δ) аппроксимация лазейки, вероятностная лазейка. Зафиксируем $\varepsilon, \delta \in (0, 1)$, множество переменных $B \subseteq V$ и алгоритм A , и рассмотрим множество означиваний \hat{B} в терминах ρ -лазейки.

— Введем на \hat{B} равномерное распределение и зададим случайную величину

$$\xi_B(\hat{b}) = \left[E[B \mid \hat{b}] \in S(A) \right].$$

Ясно, что эта величина распределена по Бернулли с $p = \rho_B$.

— Тогда B является (ε, δ) аппроксимацией лазейки, если

$$\Pr \left[1 - \varepsilon \leq \frac{1}{N} \sum_{j=1}^N \xi_j \right] \geq 1 - \delta,$$

где $N \geq \frac{4 \ln 2 / \delta}{\varepsilon^2}$. Данное условие согласно теореме Чернова TBD-REF обеспечивает тот факт, что аппроксимация ρ_B , вычисляемая по формуле

$$\hat{\rho}_B = \frac{1}{N} \sum_{j=1}^N \xi_j, \tag{1}$$

отклоняется от истинного значения ρ_B не более, чем на ε с вероятностью не менее $1 - \delta$.

1.2. Последовательные решатели

1.3. Параллельные решатели

Выводы по главе 1

ГЛАВА 2. ТЕОРЕТИЧЕСКОЕ ИССЛЕДОВАНИЕ И АРХИТЕКТУРА

2.1. Поиск вероятностных лазеек

2.1.1. Схема алгоритма

В данной главе описаны алгоритмы поиска вероятностных лазеек. Для поиска вероятностных лазеек применяются эволюционные алгоритмы, а именно, $(1 + 1)$ и $(q + h)$ алгоритмы [3], [6]. Для полного описания теоретической схемы поиска вероятностных лазеек необходимо определить особь, фитнес-функцию, а также операторы скрещивания, мутации и отбора.

Особью во всех реализованных схемах поиска вероятностных лазеек является битовая маска \overline{B} , соответствующая множеству переменных B , включенных в лазейку.

Фитнес функция. В фитнес-функции используется аппроксимация значения ρ , формально определенная уравнением 1, так как такой подход позволяет вычислять его с достаточно высокой точностью, а главное — быстро. В качестве алгоритма A используется *вывод последствий, или unit propagation (UP)*, эффективно реализованный в рамках решателя Minisat [5]. Данный алгоритм с точки зрения решателя не является полным, но имеет полиномиальное время работы.

Используется фитнес функция, описанная в TBD-REF. Её преимущество заключается в том, что она помимо максимизации $\hat{\rho}$ -значения лазейки минимизирует её размер. Максимизация $\hat{\rho}$ -значения достигается первой частью функции:

$$G_C(\overline{B}) = (1 - \hat{\rho}_B) \cdot 2^{\omega|V|},$$

где C — формула 1, V — множество переменных в формуле, $w \in (0, 1]$ — константа. Минимизация размера лазейки обеспечивается второй частью функции:

$$f_{C, \min|B|} = \hat{\rho}_B \cdot 2^{|B|}.$$

Действительно, при относительно близких значениях ρ большое влияние на функцию будет оказывать размер лазейки B . Итоговая фитнес функция выглядит следующим образом:

$$f_C(\overline{B}) = \hat{\rho}_B \cdot 2^{|B|} + (1 - \hat{\rho}_B) \cdot 2^{\omega|V|}. \quad (2)$$

Операторы В качестве основного оператора мутации был выбран зарекомендовавший себя в TBD-REF оператор Doege [11]. В процессе разработки также использовался равномерный оператор мутации, однако результаты, которые он показывал, стабильно хуже результатов Doege на всех примерах, поэтому он был отброшен, однако доступен в конфигурации. Реализованы операторы однотоочечного и двуточечного скрещивания. Генетический алгоритм $(\mu + \lambda)$ был реализован аналогично схеме, предложенной в TBD-REF.

2.1.2. Эффективный вывод последствий

Основным потребителем ресурсов в описанной схеме поиска вероятностных лазеек является алгоритм вывода последствий, реализованный в Minisat [5]. Поэтому оптимизация этого алгоритма и разработка новых подходов является крайне важной задачей для достижения высокой производительности.

Выборка. Заметим, что для достаточно маленьких B имеет смысл производить полный перебор множества \hat{B} при вычислении $\hat{\rho}$. Во-первых, это обеспечит точное значение $\hat{\rho} = \rho$, во-вторых, реализация перебора всех подстановок (то есть, перебора всех возможных назначений переменных из набора в B , что эквивалентно перебору всех чисел от 0 до $2^{|B|} - 1$ в двоичной записи) точно будет работать не медленнее, чем случайный перебор $2^{|B|}$ подстановок, так как не пользуется геренацией случайных чисел и делает строго меньше операций за счет того, что не на каждом шаге модифицируются все значения подстановки.

Данное наблюдение создает необходимость в абстракции от типа выборки. Для этого был создан интерфейс Search, а также следующие реализации:

- FullSearch — перебор всех возможных $\hat{b} \in \hat{B}$.
- RandomSearch — перебор N случайных подстановок из \hat{B} .
- UniqueSearch — перебор N уникальных случайных подстановок из \hat{B} .
- CartesianSearch — перебор всех подстановок из декартового произведения нескольких наборов подстановок, необходимая для реализации некоторых стратегий (см. Главу 2.3).

На рисунке 1 представлена схема наследования интерфейса Search. Далее в таблице 1 описаны методы интерфейса.

search

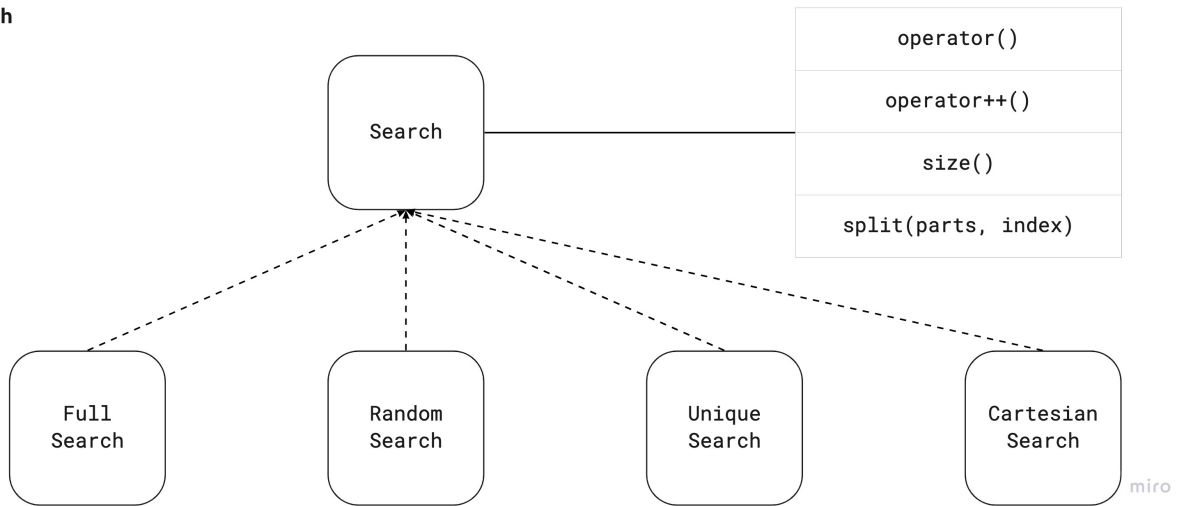


Рисунок 1 – Интерфейс Search и его реализации

Таблица 1 – Описание методов Search

Метод	Параметры	Описание
operator()	—	Возвращает текущий элемент выборки
operator++	—	Переходит на следующий элемент выборки. Возвращает false, если новый элемент – последний
size	—	Возвращает размер выборки
split	Число потоков, номер потока	Возвращает часть исходной выборки

После получения выборки необходимо вызвать метод вывода последствий на всех элементах этой выборки. Далее будут рассмотрены различные подходы к решению этой задачи: *наивный (последовательный), параллельный, вычисление ρ через дерево поиска*.

Наивный подход. Данный подход подразумевает последовательный вызов метода вывода последствий. То есть, имея выборку D и решатель A , значение ρ аппроксимируется так, как показано в листинге 1.

Листинг 1 – Наивное вычисление $\hat{\rho}$

```

function calculate_rho(A, D)
   $S \leftarrow 0$ 
  do
     $\hat{b} \leftarrow D()$ 
    if  $A(\hat{b}) = 0$  then
       $S \leftarrow S + 1$ 
    end if
  while  $D++$ 
  return  $\frac{S}{N}$ 
end function

```

Параллельная обработка выборки. Данный подход разделяет выборку на несколько выборок и обрабатывает их в разных потоках. Для этого используется метод `split`. Псевдокод данного подхода представлен в листинге 2.

Листинг 2 – Параллельное вычисление $\hat{\rho}$

```

function calculate_rho_par( $[A_i, i = \overline{1, M}]$ , D)
   $S \leftarrow 0$ 
   $[D_i] \leftarrow [D.\text{split}(M, j), j = \overline{1, M}]$ 
  return  $\sum_i^{\text{parallel}} \text{calculate\_rho}(A_i, D_i) / N$ 
end function

```

Данный подход с теоретической точки зрения крайне хорошо масштабируется, так как метод `split` занимает несравнимо мало времени по сравнению с многократным вызовом метода вывода последствий. Практическое сравнение последовательного и параллельного подходов приведено в Главе ??.

Вычисление точного значения ρ через дерево подстановок. Данный подход является новым и не имеет аналогов в известных мне решателях за ненадобностью. Однако, идея, описанная далее, является крайне эффективной оптимизацией перебора \hat{B} , как показано в Главе ??. Для вычисления точного значения ρ необходимо посчитать число подстановок $\hat{b} \in \hat{B}$, которые решаются заданным алгоритмом A . В данной работе, как упомянуто ранее, в качестве этого алгоритма используется алгоритм вывода последствий. Результатом работы этого алгоритма может быть либо информация о существовании конфликтующих подстановок переменных, либо отсутствие дополнительной информации. Заметим, что если на каком-то префиксе \hat{b} алгоритм сообщил о конфликте, то нет никакого смысла далее рассматривать подстановки с этим

префиксом, так как все они заведомо конфликтные. Можно просто прибавить к результату размер всего поддерева поиска. Иллюстрация к данной идее приведена на рисунке 2. Листинг, содержащий псевдокод данного метода, приведен в приложении (листинг ??). Данный метод был успешно реализован и встроен в исходный код решателя Minisat. Сравнительные результаты с наивным и параллельным подходами содержатся в Главе ??.

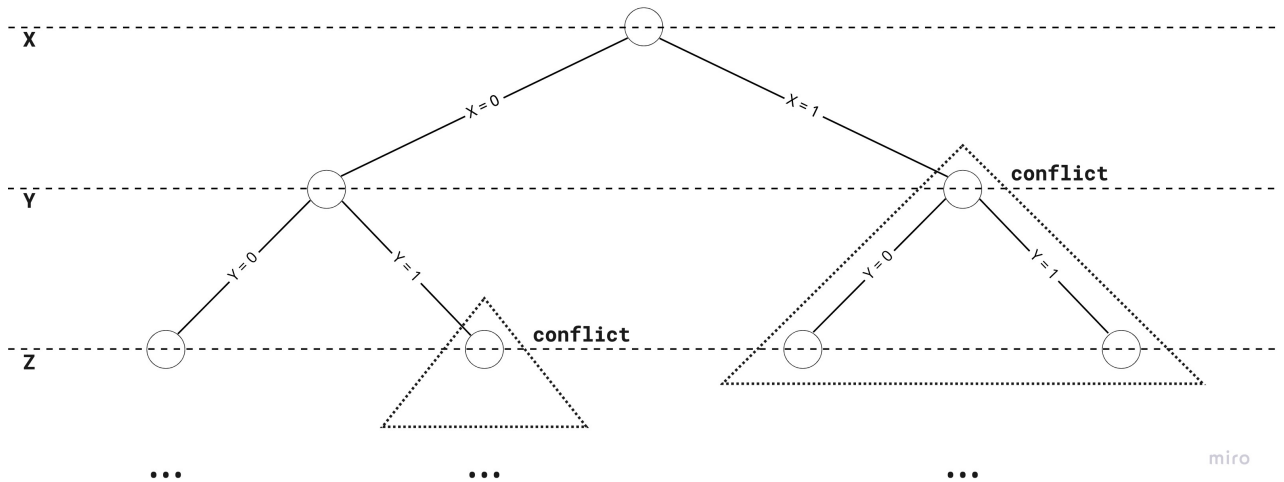


Рисунок 2 – Вычисление ρ через дерево подстановок. Здесь X, Y, Z — переменные. Подстановки $E[X \mid 1]$ и $E[\{X, Y\} \mid \{X \rightarrow 0, Y \rightarrow 1\}]$ приводят к конфликту, поэтому соответствующие поддеревья поиска можно не рассматривать.

2.2. Решение набора подстановок

В данной главе описана архитектура сервиса решения булевой функции с подстановками. Описан интерфейс сервиса, приведена схема распределения задач по последовательным решателям, описана техника обмена знаниями между решателями.

В качестве последовательных решателей используются Minisat [5], MapleCOMSPS [7], painless-maplecomsps [8]. Под последовательным имеется в виду не тип самого решателя, а невозможность параллельного решения разных задач. То есть, painless-mcomsps является параллельным решателем, но может решать лишь одну задачу одновременно. Описываемый сервис нужен именно для того, чтобы обеспечить возможность параллельного решения разных задач, что необходимо для реализации параллельных алгоритмов решения на основе подхода разделяй-и-властвуй ??.

Схема наследования интерфейса SolverService проиллюстрирована на рисунке 3. Описание методов интерфейса приведено в таблице 2.

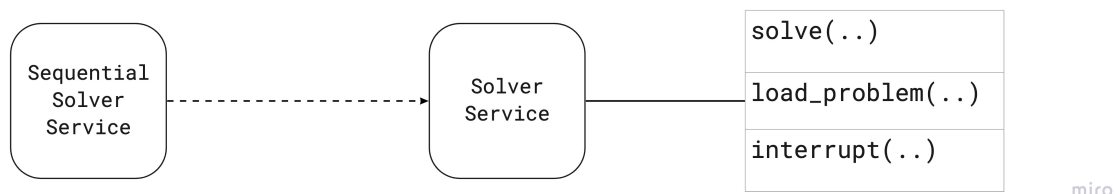


Рисунок 3 – Интерфейс SolverService и его реализация

Таблица 2 – Описание методов SolverService

Метод	Параметры	Описание
solve	Подстановка, ограничение по времени, callback-функция, вызываемая при окончании решения	Возвращает объект <code>std::future</code> типизированный результатом решения, и добавляет задачу в очередь
load_problem	—	Загружает формулу в решатель
interrupt	—	Прерывает процесс решения

Реализация `SequentialSolverService` управляет набором последовательных решателей, работающих параллельно: обеспечивает их задачами, а также, при необходимости, обеспечивает обмен знаниями. Для распределения задач реализована безопасная для использования в многопоточной среде очередь. Для обмена знаниями используются механизмы, реализованные в интерфейсе `rainless` и адаптированные для использования разных решателей и схем обмена. Схема работы `SequentialSolverService` приведена на рисунке 4.

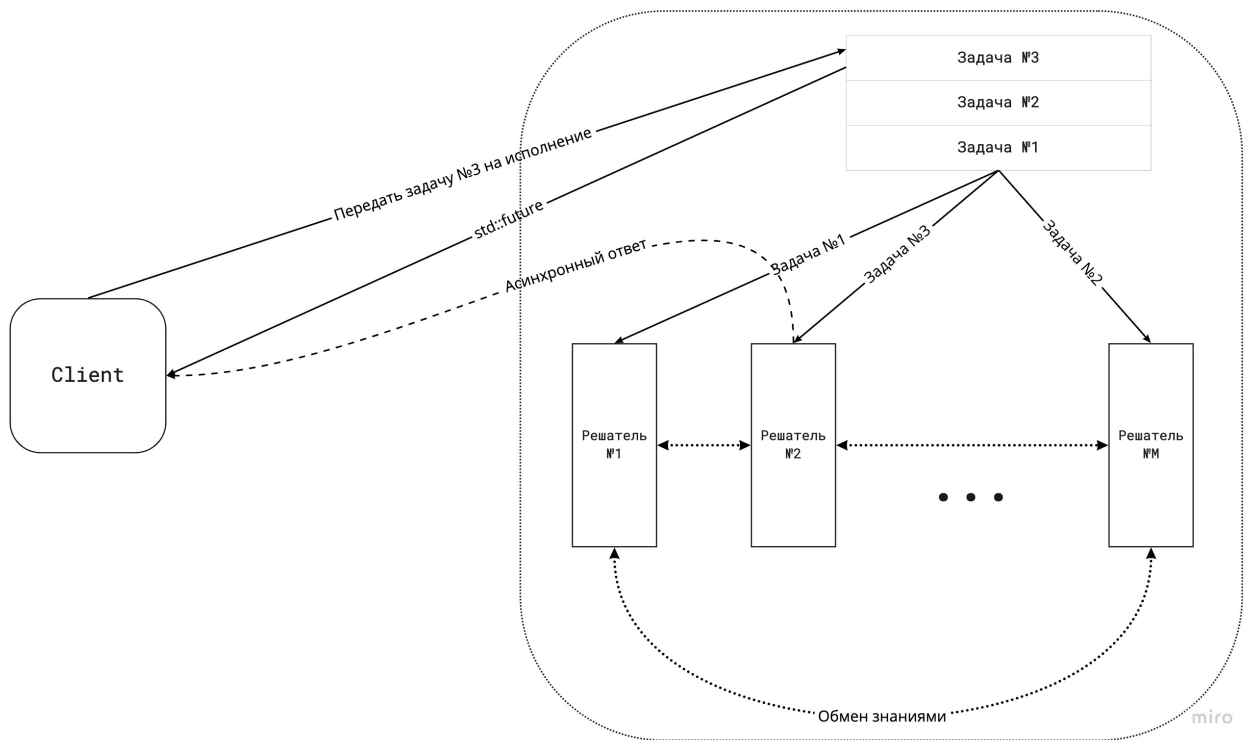


Рисунок 4 – Схема работы SequentialSolverService

2.3. Использование вероятностных лазеек

В данной главе описаны реализованные подходы к использованию вероятностных лазеек для решения задач булевой выполнимости. Описаны *прямой* и *рекурсивный* подходы к решению, а также схема с *использованием нескольких лазеек*.

2.4. Описание реализации

В данной главе описана структура исходного кода алгоритма, перечислены возможные опции как при сборке, так и при запуске приложения.

Выводы по главе 2

ГЛАВА 3. ПРАКТИЧЕСКОЕ ИССЛЕДОВАНИЕ

Выводы по главе 3

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 An Analysis of SAT-Based Model Checking Techniques in an Industrial Environment / N. Amla [и др.] // *Correct Hardware Design and Verification Methods* / под ред. D. Borriore, W. Paul. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2005. — С. 254–268. — ISBN 978-3-540-32030-2.
- 2 *Avellaneda F., Villemaire R.* Boolean Matrix Factorization with SAT and MaxSAT. — 2021. — DOI: 10.48550/ARXIV.2106.10105. — URL: <https://arxiv.org/abs/2106.10105>.
- 3 *Borisovsky P., Ereemeev A.* A Study on Performance of the (1+1)-Evolutionary Algorithm. — 2003. — ЯНВ.
- 4 *Cook S. A.* The complexity of theorem-proving procedures // IN STOC. — ACM, 1971. — С. 151–158.
- 5 *Eén N., Sörensson N.* An Extensible SAT-solver // *Theory and Applications of Satisfiability Testing* / под ред. E. Giunchiglia, A. Tacchella. — Berlin, Heidelberg : Springer Berlin Heidelberg, 2004. — С. 502–518. — ISBN 978-3-540-24605-3.
- 6 *Lingaraj H.* A Study on Genetic Algorithm and its Applications // *International Journal of Computer Sciences and Engineering*. — 2016. — Окт. — Т. 4. — С. 139–143.
- 7 MapleCOMSPS, MapleCOMSPS_LRB, MapleCOMSPS_CHB [Электронный ресурс] / J. H. Liang [и др.]. — 2016. — URL: <https://docs.google.com/a/gsd.uwaterloo.ca/viewer?a=v%5C&pid=sites%5C&srcid=Z3NkLnV3YXRlcmxvby5jYXxtYXBsZXNhZHxneDo2YWEzYjEzN2JmY2I1Y> (дата обр. 07.04.2022).
- 8 PaInleSS: a Framework for Parallel SAT Solving. / L. Le Frioux [и др.] // *The 20th International Conference on Theory and Applications of Satisfiability Testing*. T. 10491. — Melbourne, Australia : Springer, 08.2017. — С. 233–250. — (Lecture Notes in Computer Science). — DOI: 10.1007/978-3-319-66263-3_15. — URL: <https://hal.archives-ouvertes.fr/hal-01540785>.

- 9 *Patsakis C.* RSA private key reconstruction from random bits using SAT solvers // IACR Cryptol. ePrint Arch. — 2013. — Т. 2013. — С. 26.
- 10 Solving Open Job-Shop Scheduling Problems by SAT Encoding / M. KOSHIMURA [и др.] // IEICE Transactions on Information and Systems. — 2010. — Т. E93.D, № 8. — С. 2316–2318. — DOI: 10.1587/transinf.E93.D.2316.
- 11 *Doerr B., Doerr C.* Optimal Parameter Choices Through Self-Adjustment: Applying the 1/5-th Rule in Discrete Settings [Электронный ресурс]. — 2015. — URL: <http://arxiv.org/abs/1504.03212>.