

Министерство образования и науки Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ”

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

*СОЛВЕР ДЛЯ ЗАДАЧИ ЛИНЕЙНОЙ ОПТИМИЗАЦИИ В  
ПРОСТРАНСТВЕ СБАЛАНСИРОВАННЫХ ПОТОКОВ С  
ТЕРМОДИНАМИЧЕСКИМИ ОГРАНИЧЕНИЯМИ*

Автор Якутов Дмитрий Алексеевич

Направление подготовки 01.03.02 Прикладная математика  
и информатика

Квалификация Бакалавр

Руководитель Сергушичев А.А., к.т.н.

**К защите допустить**

Зав. кафедрой КТ Васильев В.Н., проф., д.т.н.

“ ” \_\_\_\_\_ 2018 г.

Санкт-Петербург, 2018 г.

Студент Якутов Д.А. Группа М3439 Кафедра КТ Факультет ИТиП

Направленность (профиль), специализация

*Математические модели и алгоритмы в разработке программного обеспечения*

ВКР принята «\_\_\_\_» \_\_\_\_\_ 2018 г.

Оригинальность ВКР \_\_\_\_\_ %

ВКР выполнена с оценкой \_\_\_\_\_

Дата защиты «\_\_\_\_» июня 2018 г.

Секретарь ГЭК Павлова О.Н. \_\_\_\_\_

Листов хранения \_\_\_\_\_

Демонстрационных материалов/Чертежей хранения \_\_\_\_\_ *отсутствуют*

**Министерство образования и науки Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**  
**УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,**  
**МЕХАНИКИ И ОПТИКИ”**

**УТВЕРЖДАЮ**

Зав. кафедрой *КТ* \_\_\_\_\_

*проф. Васильев В. Н.* \_\_\_\_\_

\_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**  
**НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

Студенту *Якутову Д.А.* \_\_\_\_\_ Группа *М3439* Кафедра *КТ* Факультет *ИТиП*

Руководитель *Сергушичев А.А., кандидат технических наук, доцент, Университет ИТМО* \_\_\_\_\_

**1 Наименование темы:** *Солвер для задачи линейной оптимизации в пространстве сбалансированных метаболических потоков с термодинамическими ограничениями* \_\_\_\_\_

**Направление подготовки (специальность)** *01.03.02 Прикладная математика и информатика* \_\_\_\_\_

**Квалификация** \_\_\_\_\_ *Бакалавр* \_\_\_\_\_

**2. Срок сдачи студентом законченной работы** \_\_\_\_\_ *15 мая 2018 г.* \_\_\_\_\_

**3. Техническое задание и исходные данные к диссертации**

*1 Изучение линейного программирования*

*2 Изучение пакета программного обеспечения CPLEX*

*3 Изучение существующих решений задачи*

*4 Изучение данных инстансов на выявление общих свойств*

*5 Разработка идей решения задачи*

*6 Реализация разработанных решений и решений, найденных ранее*

*7 Сравнение реализованных решений*

#### **4 Содержание выпускной работы (перечень подлежащих разработке вопросов)**

*1 Описание областей линейного программирования и анализа баланса потоков*

*2 Постановка конкретной задачи и изучение существующих решений*

*3 Описание разработанных идей решений и их программной реализации*

*4 Сравнение найденных решений с существующими решениями*

#### **5 Перечень графического материала (с указанием обязательного материала)**

*Графические материалы и чертежи работой не предусмотрены*

#### **6 Исходные материалы и пособия**

*Опубликованные статьи на тему «Анализ баланса потоков»*

**7 Дата выдачи задания** «21» «ноября» 2017г.

Руководитель \_\_\_\_\_

Задание принял к исполнению \_\_\_\_\_

«21» «ноября» 2017г.

**Министерство образования и науки Российской Федерации**  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**“САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ”**

## АННОТАЦИЯ

### ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

Студента Якутова Дмитрия Алексеевича

Наименование темы ВКР: Солвер для задачи линейной оптимизации в пространстве сбалансированных метаболических потоков с термодинамическими ограничениями

Наименование организации, где выполнена ВКР Университет ИТМО

#### ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

1 Цель исследования: Разработать практический солвер для задачи линейной оптимизации в пространстве сбалансированных метаболических потоков, учитывающий законы термодинамики

2 Задачи, решаемые в ВКР Изучение областей линейного программирования и анализа баланса потоков, разработка новых решений поставленной задачи, оценка эффективности их работы, сравнение их с существующими решениями

3 Число источников, использованных при составлении обзора 3

4 Полное число источников, использованных в работе 6

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 |
| 0               | 0              | 0            | 6               | 0              | 0            |

6 Использование информационных ресурсов Internet да, число ресурсов: 5

7 Использование современных пакетов компьютерных программ и технологий

| Пакеты компьютерных программ и технологий | Параграф работы                |
|---|--------------------------------|
| Java 1.8.0_152                            | Разработка и сравнение решений |
| CPLEX 12.7                                | Разработка решений             |
| Efficient Java Matrix Library             | Разработка решений             |
| Cytoscape 3.6                             | Изучение данных                |

8 Краткая характеристика полученных результатов

*Разработанные решения превосходят по эффективности существующие решения на тестах, предоставленных лабораторией Университета ИТМО. Данные решения можно использовать на практике для анализа метаболических потоков.*

9 Полученные гранты, при выполнении работы: нет

10 Наличие публикаций и выступлений на конференциях по теме выпускной работы: нет

Студент Якутов Д.А. \_\_\_\_\_

Руководитель Сергушичев А.А. \_\_\_\_\_

“ \_\_\_\_\_ ” \_\_\_\_\_ 2018 г.

## Оглавление

|   |    |
|---|----|
| Введение .....  | 5  |
| ГЛАВА 1. ОБЗОР БИОИНФОРМАТИКИ И ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ .....              | 6  |
| 1.1 Термины .....   | 6  |
| 1.1.1 Линейное программирование .....   | 6  |
| 1.1.2 Метаболические потоки .....   | 7  |
| 1.2 Задачи линейной оптимизации сбалансированных метаболических потоков ..... | 9  |
| 1.3 Обзор библиотеки CPLEX .....  | 10 |
| 1.4 Обзор существующего решения .....   | 11 |
| 1.5 Цели работы.....  | 12 |
| 1.6 Выводы по главе 1 .....   | 12 |
| ГЛАВА 2. ПРЕДЛАГАЕМЫЕ МЕТОДЫ РЕШЕНИЯ.....                                     | 13 |
| 2.1 Солверы поставленной задачи линейной оптимизации .....                    | 13 |
| 2.1.1 Солвер SimpleSolver.....  | 13 |
| 2.1.2 Солвер PathSolver .....   | 17 |
| 2.1.3 Солвер NullSpaceSolver.....   | 20 |
| 2.1.4 Солвер LinearSpaceDecomposer.....                                       | 23 |
| 2.1.5 Солвер ClosedSolver.....  | 21 |
| 2.2 Солверы вспомогательных задач линейной оптимизации .....                  | 21 |
| 2.2.1 Солвер CycleSolver .....  | 21 |
| 2.2.2 Солвер ContinuousSplitter .....   | 22 |
| 2.3 Выводы по главе 2 .....   | 23 |
| ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ .....                                 | 25 |
| 3.1 Описание компьютера и программного обеспечения.....                       | 25 |
| 3.2 Проверка солверов на маленьком инстансе .....                             | 25 |
| 3.3 Проверка солверов на практических инстансах .....                         | 27 |
| 3.3.1 Сбор данных об инстансах.....   | 27 |
| 3.3.2 Тестирование решений на практических инстансах .....                    | 30 |
| 3.4 Сравнение результатов .....   | 34 |
| 3.5 Выводы по главе 3 .....   | 34 |
| ВЫВОДЫ .....  | 35 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....   | 36 |

## Введение

Изучение метаболизма является актуальной областью биологии. Метаболизм играет большую роль в биологических процессах, в частности в раковых клетках и иммунной системе. Моделирование метаболизма позволяет изучать и предсказывать влияние внешних воздействий на биологические процессы в организме. В настоящий момент появилась возможность получения данных для изучения, использующая меньше ресурсов, чем раньше. Поэтому анализ получения одних метаболитов из других с помощью известных на текущий момент реакций является актуальной темой на сегодняшний день.

Большинство из существующих решений задач, схожих с задачей, решаемой в этой работе, учитывают закон сохранения масс, но не учитывают законы термодинамики: в получившихся решениях присутствуют положительные круговые потоки. Задача, поставленной в данной работе, является NP-полной, поэтому она сильно труднее, чем задачи, не учитывающие термодинамические ограничения.

В главе 1 данной работы рассказаны основные определения, необходимые для понимания дальнейших рассуждений, поставлена решаемая задача и рассказано о том, какое существует решение на данный момент.

В главе 2 работы описаны предложенные сведения задачи к задаче целочисленного линейного программирования и алгоритмы решения задачи на основе этих сведений. Также в главе 2 решались возникающие по ходу решения побочные задачи и доказывались нетривиальные утверждения, используемые в процессе решения.

В главе 3 приведены экспериментальные результаты работы солверов, реализованных по описанию в главе 2, а также объяснены получившиеся результаты.



## ГЛАВА 1. ОБЗОР БИОИНФОРМАТИКИ И ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В этой главе рассмотрены термины, использующиеся в предметной области и данной работе, описана задача, решаемая в ходе данной работы, описаны методы, использованные для решения задачи. Показано существующее решение, проведен его анализ, выявлены его достоинства и недостатки.

### 1.1 Термины

#### 1.1.1 Линейное программирование

*Определение 1.*  $\mathbb{R}$  — множество вещественных (действительных) чисел.

*Определение 2.*  $\mathbb{R}_+$  — множество неотрицательных вещественных чисел.

*Определение 3.* Задача линейного программирования (LP) — задача нахождения множества переменных  $x_1, x_2, \dots, x_n \in \mathbb{R}_+$ , удовлетворяющих  $m$  условиям ( $j = 1 \dots m$ ):

$$\sum_{i=1}^n a_j^i \cdot x_i \leq b_j \quad (j)$$

При этом среди всех решений данной системы неравенств находится решение, максимизирующее значение целевой функции:

$$\sum_{i=1}^n c_i \cdot x_i \rightarrow \max$$

Задача линейного программирования хорошо исследована и имеет полиномиальное решение от переменных  $n$  и  $m$ .

Несмотря на большое количество ограничений в формулировке задачи линейного программирования, большинство из них можно обойти. Например, если для решения задачи нужна переменная  $x$  не обязательно неотрицательная, то вместо нее можно ввести две неотрицательные переменные  $x_+$  и  $x_-$  и использовать разность  $x_+ - x_-$  в качестве переменной  $x$ . Также можно не максимизировать значение целевой функции, а минимизировать его,

записывать ограничения, использующие знак неравенства в другую сторону или знак равенства.

*Определение 4.* Задача целочисленного линейного программирования (ILP) — задача, схожая с задачей линейного программирования, отличающаяся от нее тем, что некоторые из переменных  $x_1, \dots, x_n$  обязаны быть целыми. Данная задача существенно сложнее задачи линейного программирования, так как она является NP-трудной, то есть любую задачу из класса NP можно свести к решению этой задачи. На данный момент нет известного способа решить ее, используя время, пропорциональное некоторому полиному, зависящему от переменных  $n$  и  $m$ .

*Определение 5.* Метод ветвей и границ — один из методов для нахождения решений задач оптимизации, в частности, задачи целочисленного линейного программирования. Данный метод используется в данной работе для решения возникающих задач. Подробности метода не являются необходимостью для понимания данной работы.

*Определение 6.* Солвер — программа, позволяющая по входным данным найти решение фиксированной задачи оптимизации.

*Определение 7.* Ядро линейного оператора  $A$  — множество векторов  $x$  таких, что  $A \cdot x = 0$ . Как известно из линейной алгебры, ядро линейного оператора является линейным пространством.

### 1.1.2 Метаболические потоки

*Определение 8.* Метаболиты — продукты метаболизма, некоторые вещества.

*Определение 9.* Внешний метаболит — метаболит, потребляемый или выделяемый рассматриваемый организмом.

*Определение 10.* Внутренний метаболит — метаболит, не являющийся внешним, то есть вещество, являющееся внутренним продуктом метаболизма.

*Определение 11.* Реакция — процесс превращения некоторого набора веществ в другой набор веществ. В данной работе под реакцией

подразумевается набор метаболитов и соотношение, в котором реакция потребляет и продуцирует метаболиты.

*Определение 12.* Внешняя реакция — реакция, использующая или продуцирующая внешний метаболит.

*Определение 13.* Внутренняя реакция — реакция, не являющаяся внешней. Такая реакция использует только внутренние вещества.

*Определение 14.* Стехиометрическая матрица ( $S$ ) — матрица размерности  $m \times n$ , где  $m$  — количество внутренних метаболитов в системе, а  $n$  — количество реакций в системе. Эта матрица описывает соотношения между веществами и реакциями. В клетке  $(i, j)$  матрицы стоит стехиометрический коэффициент метаболита  $i$  в реакции  $j$ . Этот коэффициент равен 0, если метаболит не участвует в реакции, положительный, если метаболит генерируется в результате реакции, и отрицательный, если метаболит потребляется в результате реакции. В зафиксированном столбце  $j$  эти коэффициенты выражают пропорцию, в которой метаболиты потребляются и продуцируются в процессе реакции  $j$ .

*Определение 15.* Вектор потоков — вектор неотрицательных вещественных чисел  $v$  длины  $n$ , выражающий величины потоков по всем реакциям в системе.

*Определение 16.* Ограничения баланса — ограничения, выражающие правило сохранения масс: в каждый момент времени каждое вещество генерируется в той же мере, что потребляется. Это правило можно выразить матричным уравнением:

$$S \cdot v = 0$$

*Определение 17.* Анализ баланса потоков (FBA) — метод математического моделирования метаболизма, позволяющий определить скорость реакций в метаболической сети [1]. Этот метод учитывает ограничения баланса и позволяет изучать и предсказывать поведение организма при влиянии внешних воздействий.

*Определение 18.*  $S_{int}$  — сужение матрицы  $S$  на внутренние реакции.

*Определение 19.*  $N_{int}$  — матрица, описывающая базис ядра линейного оператора, описанного матрицей  $S_{int}$ .

*Определение 20.* Круговой поток — вектор потоков, в котором потоки по всем внешним реакциям равны нулю.

*Определение 21.* Положительный круговой поток — круговой поток, имеющий, по крайней мере, одну реакцию с положительным потоком по ней. Такие потоки являются некорректными с термодинамической точки зрения: закон сохранения энергии не выполняется для таких потоков.

*Определение 22.* Подпоток — поток  $v_1$  является подпотокотом потока  $v_2$ , если  $\forall i: v_{1,i} \leq v_{2,i}$ . В частности, как видно из определений 17 и 18, любой подпоток положительного кругового потока также является круговым подпотокотом.

*Определение 23.* Термодинамические ограничения — ограничения, описывающие отсутствие у заданного потока положительного кругового подпотокотом.

*Определение 24.* Корректный поток — поток  $v$ , удовлетворяющий ограничениям баланса и термодинамическим ограничениям.

*Определение 25.* Вектор весов — вектор  $w \in \mathbb{R}^n$ .

*Определение 26.* Инстанс — входные данные, выражающиеся данными стехиометрической матрицей, вектором весов, характеризующим целевую функцию, векторами нижних и верхних границ для каждой реакции.

## 1.2 Задачи линейной оптимизации сбалансированных метаболических потоков

Задача, рассмотренная в данной работе, заключается в нахождении по данному инстансу корректного потока, максимизирующего значение целевой функции:

$$(v, w) = \sum_{i=1}^n w_i v_i \rightarrow \max \quad (1)$$

Также для каждой реакции установлены нижнее и верхнее ограничения на величину потока через нее:

$$0 \leq v_i \leq ub_i \quad (2)$$

В частности, в работе уделено внимание особым целевым функциям, вектора весов которых состоят из нулей и одной единицы. В такой задаче требуется максимизировать поток по одной фиксированной реакции.

Ограничения баланса записаны в терминах ограничений задачи линейного программирования. Максимизируемая целевая функция также записана в терминах задачи линейного программирования. Однако ограничения на корректность потока не записываются так просто. Задача данной работы заключается в том, чтобы разработать запись термодинамических ограничений в терминах целочисленного линейного программирования так, чтобы программная реализация решения была как можно более эффективной.

### 1.3 Обзор библиотеки CPLEX

В процессе изучения CPLEX были изучены подробности метода ветвей и границ. Процесс нахождения решения в этом методе заключается в том, чтобы поддерживать верхнюю границу целевой функции, то есть ее максимально возможное значение, и разбиваться на разные случаи. Разбиваться имеет смысл, например, в следующих ситуациях:

- если в этих случаях имеется возможность уточнить верхнюю границу, например, решив задачу без ограничений целочисленности некоторых переменных;
- если в этих случаях можно уменьшить размерность задачи путем добавления новых ограничений.

Таким образом, структуру решения задачи методом ветвей и границ можно представить корневым деревом, каждой вершине которого сопоставляется некоторое уточнение данной задачи: локальные дополнительные ограничения и локальная верхняя граница значения целевой функции.

Для общения с пользователем CPLEX предоставляет иерархию классов Callback, позволяющую добавлять ограничения в процессе разбиения вершины на несколько. Мы будем использовать следующие из них:

- IncumbentCallback. Этот класс позволяет по вектору, найденному в текущей ветке решения, сказать методу, что это решение не является корректным, и изменить состояние солвера.
- LazyConstraintCallback. Этот класс позволяет добавить новые ограничения в задачу, причем как глобально во все ветви метода ветвей и границ, так и локально: в этом случае добавленные ограничения будут действовать только в текущей вершине и всех вершинах ее поддерева. В процессе решения мы будем использовать добавление глобальных ограничений, так как они корректны во всех ветках метода.

#### 1.4 Обзор существующего решения

На данный момент существует следующий метод решения задачи, описанный в статье [2]. Для реализации решения вводятся дополнительные переменные для каждой реакции:  $a_i$  — целочисленная переменная, принимающая значения 0 или 1, описывающая, является ли реакция активной, то есть правда ли, что по ней может идти ненулевой поток:

$$0 \leq a_i \leq 1 \quad (3)$$

$$0 \leq v_i \leq M \cdot a_i \quad (4)$$

Здесь и далее  $M$  — некоторое положительное число, заведомо большее величины потока через любую реакцию.

Также для описания ограничения на отсутствие положительного кругового потока вводится по одной дополнительной переменной  $G_i$  на каждую реакцию. В статье [2] доказано, что для отсутствия кругового потока достаточно ввести следующие ограничения:

$$-(M + 1) \cdot a_i + 1 \leq G_i \leq M - (M + 1) \cdot a_i \quad (5)$$

$$N_{int} \cdot G = 0 \quad (6)$$

### **1.5 Цели работы**

Основной целью данной работы являются разработка и реализация решения, эффективно работающего на инстансах, которые бывают на практике.

В ходе данной работы решались следующие задачи:

- Обработка данных стехиометрических матриц практических инстансов для выявления общих свойств;
- Изучение возможностей пакета программного обеспечения CPLEX [3], предназначенного для решения задач оптимизации, в частности, задачи целочисленного линейного программирования;
- Разработка сведения задачи к задаче целочисленного линейного программирования;
- Реализация наиболее эффективных решений и сравнение их с существующим решением.

### **1.6 Выводы по главе 1**

В главе 1 были даны определения из предметных областей, связанных с данной работой: линейное программирование, биоинформатика. Была поставлена решаемая задача и описано существующее решение.

## ГЛАВА 2. ПРЕДЛАГАЕМЫЕ МЕТОДЫ РЕШЕНИЯ

В данной главе описываются найденные решения и доказываются нетривиальные факты, относящиеся к сведению задачи к задаче линейного программирования.

### 2.1 Солверы поставленной задачи линейной оптимизации

В этом разделе описываются солверы поставленной в обзоре задачи. Солвер SimpleSolver реализует «ленивое» добавление ограничений на круговые потоки с помощью интерфейса Callback библиотеки CPLEX. Солвер PathSolver использует новый вид ограничений, базирующихся на большом количестве круговых потоков, отличающихся друг от друга небольшим числом активных реакций. Солвер NullSpaceSolver развивает идею солвера PathSolver, минимизируя размер ограничения. Солвер ClosedSolver реализован на основе статьи [2].

#### 2.1.1 Солвер SimpleSolver

Идея солвера SimpleSolver базируется на следующем сведении:

$$(w, v) \rightarrow \max \quad (1)$$

$$0 \leq v_i \leq r b_i \quad (2)$$

$$S \cdot v = 0 \quad (3)$$

$$a_i \in \{0, 1\} \quad (4)$$

$$v_i \leq M \cdot a_i \quad (5)$$

$$(c, a) + 1 \leq \sum_{i=1}^n c_i, \quad c \in \mathcal{C} \quad (6)$$

Как и в решении [2], здесь используется вектор дополнительных переменных  $a_i$ , описывающих активность реакций. Ограничение (5) говорит о том, что по реакции не может идти поток, если она не является активной с точки зрения переменной  $a_i$ .

Здесь и далее  $\mathcal{C}$  — множество векторов следующей структуры. Вектор  $c \in \{0, 1\}^n$  попадет в множество  $\mathcal{C}$ , если и только если существует положительный круговой поток  $v'$  такой, что  $c_i = \text{signum}(v'_i)$  для всех  $i$ .



Векторы этого множества описывают наборы реакций, по которым можно пустить некоторый положительный поток. Ограничение (6) говорит о том, что среди каждого такого набора реакций найдется реакция, не являющаяся активной. Это гарантирует нам отсутствие кругового потока по данному набору реакций, а так как множество  $\mathcal{C}$  содержит все такие наборы, то в нашем потоке не будет положительного кругового подпотока, то есть все решения этой системы являются корректными. С другой стороны, может быть ситуация, когда какая-то реакция является активной с точки зрения переменной  $a_i$ , но при этом поток по ней нулевой: у нас нет ограничений на то, что поток обязан быть ненулевым в случае активности реакции. Тогда одно из ограничений (6) может не выполняться, но положительного кругового подпотока может и не быть. В этом случае можно заметить, что солверу ничто не мешает сделать эту реакцию неактивной: от этого не сломается ограничение (5), а в ограничениях (6) левые части неравенств только уменьшатся.

Основной проблемой такого сведения является экспоненциальный размер множества  $\mathcal{C}$ , что делает недопустимым использование такого сведения на практике. В решении данной проблемы нам поможет иерархия классов Callback из CPLEX. Воспользуемся следующим алгоритмом:

1. Создадим экземпляр задачи целочисленного линейного программирования, в который занесем информацию о переменных  $v_i$ , и  $a_i$ , целевой функции (1) и ограничениях (2) – (5). Будем поддерживать текущее множество  $\mathcal{C}$ . Изначально оно будет пустое.
2. Решим текущую задачу линейного программирования. Мы получим конкретные значения  $v_i$  и  $a_i$ . С помощью солвера CycleSolver, о котором сказано в разделе 2.2.1, найдем в этом решении наибольший положительный круговой подпоток или определим, что такового нет.
  - 2.1. Если такового нет, то текущая ветвь метода ветвей и границ завершена, ее решение найдено.

2.2. Пусть нашелся положительный круговой подпоток  $v'$ . Найдем в потоке  $v'$  множество реакций с положительным потоком по ним, создаем соответствующий вектор  $c$  и добавляем соответствующее ограничение (6), пополняя множество  $C$  этим вектором. Так как  $v'$  является подпоток  $v$ , то все неактивные реакции  $v$  также являются неактивными в  $v'$ , а значит, если  $v$  удовлетворяет всем предыдущим ограничениям (6), то и  $v'$  им удовлетворяет. А если так, то новое добавленное ограничение сужает множество корректных потоков. Теперь все потоки, имеющие  $v'$  в качестве подпотока, будут некорректными, в то время как до этого некоторые из них были корректными: например,  $v$  и  $v'$ .

К сожалению, такой солвер добавляет очень большое количество ограничений в процессе работы, так что реализация такого солвера все еще непригодна для использования в практических целях.

В процессе обработки положительных круговых подпотоков было выяснено, что многие из найденных подпотоков имеют общие круговые подпотоки. Это означает, что вместо блокировки этих потоков было бы достаточно заблокировать их более маленький общий подпоток. На рисунке 1 представлена стехиометрическая матрица некоторого инстанса в виде графа. Вершины графа соответствуют веществам, а ориентированные ребра соответствуют реакциям, которые из одного вещества получают другое с некоторым коэффициентом.

В этом инстансе существуют положительные круговые потоки для активных множеств реакций  $\{R_1, R_5, R_6\}$ ,  $\{R_2, R_5, R_6\}$ ,  $\{R_1, R_2, R_5, R_6\}$ ,  $\{R_1, R_3, R_4\}$ ,  $\{R_2, R_3, R_4\}$ ,  $\{R_1, R_2, R_3, R_4\}$ ,  $\{R_1, R_3, R_4, R_5, R_6\}$ ,  $\{R_2, R_3, R_4, R_5, R_6\}$ ,  $\{R_1, R_2, R_3, R_4, R_5, R_6\}$ . Для каждого из этих множеств реакций должно быть верно, что хотя бы одна реакция в нем должна быть неактивной. Однако не обязательно блокировать каждый из них, добавляя ограничение (6). Например, если множество  $\{R_1, R_5, R_6\}$  содержит неактивную реакцию, то и все его надмножества, включая, например,  $\{R_1, R_3, R_4, R_5, R_6\}$ , содержат неактивную

реакцию. Это значит, что нет нужды добавлять ограничение (6) для круговых потоков, для которых мы можем найти собственное подмножество реакций, для которого также существует положительный круговой поток, состоящий только из этих реакций.

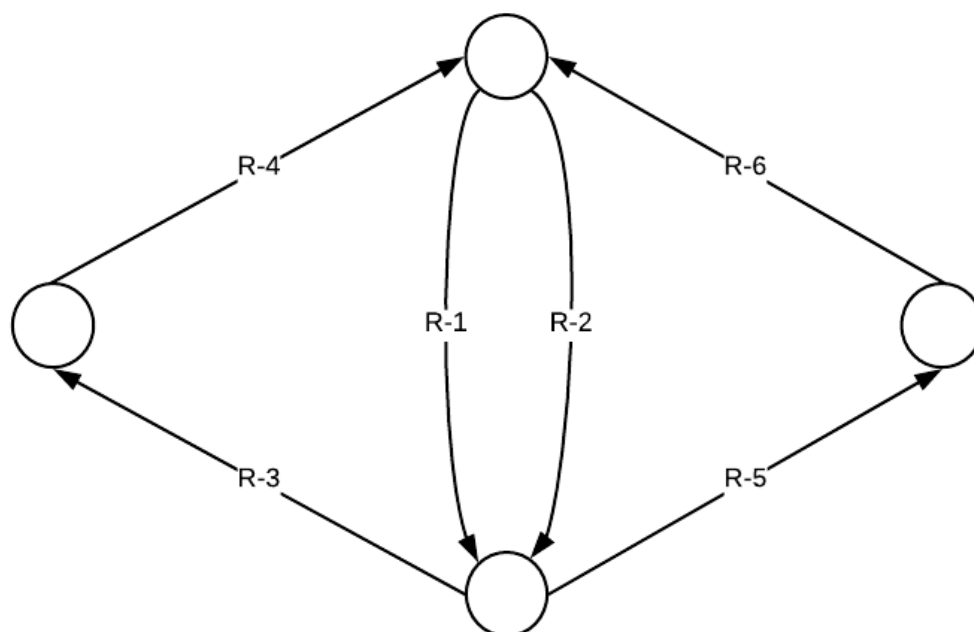


Рисунок 1 — Пример инстанса с большим количеством циклов

Мы пришли к следующей задаче: по данному круговому потоку следует найти его как можно меньший подпоток и добавить его в качестве ограничения. Выше было доказано, что CycleSolver находит поток, добавляя который в ограничения (6), мы сужаем множество допустимых решений. Отношение «быть подпотоком» является транзитивным: более маленьким подпотоком мы заблокируем то же множество потоков, а возможно и еще какие-то. Значит, наименьшим положительным круговым подпотоком потока, полученного с помощью солвера CycleSolver, мы заблокируем непустое множество новых потоков, то есть добавление такого ограничения имеет смысл.

Таким образом, мы изменим пункт 2.2 алгоритма следующим образом:

2.2.1 Если положительный круговой подпоток нашелся, найдем его наименьший положительный круговой подпоток, воспользовавшись

солвером ContinuousSplitter, сведение к задаче линейного программирования которого описано в разделе 2.2.2. Создадим вектор  $s$  из реакций, которые имеют ненулевой поток в нем и добавим соответствующее ограничение (6), пополняя множество  $\mathcal{C}$ .

Реализацию солвера SimpleSolver можно найти в пакете `ru.ifmo.cycles.solver` исходного кода.

### 2.1.2 Солвер PathSolver

Солвер SimpleSolver подходит для решения поставленной задачи, но у него есть свои проблемы. В инстансах, на которых тестировались решения, даже маленьких циклов, найденных с помощью солвера ContinuousSolver, может быть очень много. Например, на рисунке 2, множества реакций  $\{R_1, R_4, R_6, R_8, R_9, R_{10}\}$ ,  $\{R_2, R_5, R_6, R_8, R_9, R_{10}\}$ ,  $\{R_3, R_5, R_7, R_8, R_9, R_{10}\}$  и другие являются минимальными: нельзя выбрать непустое собственное подмножество этих реакций таких, что существует положительный круговой поток через эти реакции.

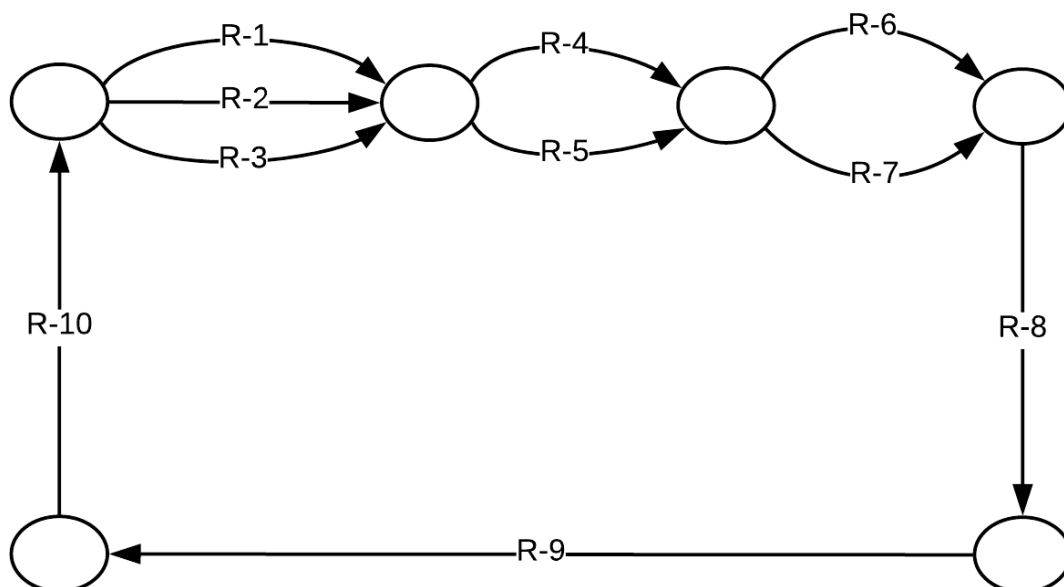


Рисунок 2 — Пример инстанса с большим количеством минимальных циклов

Пример на рисунке 2 иллюстрирует то, как множество минимальных циклов может быть экспоненциального размера. Солвер PathSolver использует идею, как уменьшить количество ограничений на подобных примерах.

Допустим, мы уже заблокировали круговой поток  $\{R_1, R_4, R_6, R_8, R_9, R_{10}\}$ . Посмотрим, как отличаются потоки  $\{R_1, R_4, R_6, R_8, R_9, R_{10}\}$  и  $\{R_2, R_4, R_6, R_8, R_9, R_{10}\}$ . Отличаются они тем, что во втором круговом потоке есть реакция  $R_2$ , но нет реакции  $R_1$ . Хочется как-то записать ограничения, которые описывают, как вместо одной реакции использовать другие: «В любом потоке, где используется реакция  $R_1$ , можно вместо нее использовать реакцию  $R_2$ ». Аналогично вместо реакции  $R_1$  можно использовать реакцию  $R_3$ , вместо реакции  $R_4$  — реакцию  $R_5$ , а вместо реакции  $R_6$  — реакцию  $R_7$ . Таким образом, вместо того, чтобы блокировать все 12 круговых потоков, можно заблокировать один из них и описать, как использовать 4 «обходных пути».

Формализуем, как записать такие ограничения. Пусть есть три дизъюнктивных множества реакций:  $A$ ,  $B$ ,  $C$ . При этом  $C$  не пусто, и из реакций множества  $A \cup C$  можно построить положительный круговой поток, и из реакций множества  $B \cup C$  можно построить положительный круговой поток. При этом круговые потоки из реакций  $A \cup C$  уже заблокированы. Теперь мы хотим заблокировать круговые потоки из реакций  $B \cup C$ . Так как круговые потоки из реакций  $A \cup C$  уже заблокированы, то либо среди реакций множества  $C$  есть неактивная, а тогда круговые потоки из реакций множества  $B \cup C$  тоже заблокированы, либо среди реакций множества  $A$  есть неактивная, и в этом случае мы хотим, чтобы среди реакций множества  $B$  тоже была неактивная. Сформулируем это следующим образом: «Все реакции множества  $A$  активны тогда и только тогда, когда все реакции множества  $B$  активны». Для реализации этой формулировки добавим новую вещественную переменную  $p$ , привязанную к этой тройке дизъюнктивных множеств:

$$0 \leq p \leq M \quad (7)$$

$$p + |A| - 1 \geq \sum_{i \in A} a_i \quad (8)$$

$$a_i \geq p, \quad i \in A \quad (9)$$

$$p + |B| - 1 \geq \sum_{i \in B} a_i \quad (10)$$

$$a_i \geq p, \quad i \in B \quad (11)$$

Если значение переменной  $p$  будет строго больше 0, то из ограничений (9) и (11) будет следовать, что все реакции из множеств  $A$  и  $B$  активны. А если значение переменной  $p$  равно 0, то ограничение (8) примет вид:

$$\sum_{i \in A} a_i \leq |A| - 1 \quad (12)$$

Ограничение (12) совпадает с ограничением (6). В этом случае хотя бы одна из переменных в сумме слева будет иметь значение 0, то есть не все реакции из множества  $A$  активны. Аналогично, в случае, когда значение переменной  $p$  равно 0, из ограничения (10) будет следовать, что не все реакции из множества  $B$  активны. Таким образом, эти неравенства позволяют нам записать ограничения вида «Активность всех реакций из  $A$  равносильна активности всех реакций из  $B$ » и описать, как вместо одного пути в круговом потоке использовать другой.

Также заметим, что эти ограничения не зависят от множества  $C$ , то есть размеры этих ограничений зависят от того, как отличаются два пересекающихся круговых потока, а от их размера не зависят.

Итоговый алгоритм выглядит следующим образом:

- Записываем ограничения (1) – (5), такие же, как для солвера SimpleSolver;
- В IncumbentCallback нам придет решение, которое является корректным с точки зрения текущего множества ограничений. Как и в случае с SimpleSolver, найдем в нем наименьший по числу реакций круговой поток  $C$ .
- Будем поддерживать список уже найденных ранее циклов *savedCycles*. Найдем среди круговых потоков в этом списке такой круговой поток  $C_1$ , что он пересекается с круговым потоком  $C$  по реакциям. Если таких

потоков несколько, выберем тот, пересечение с которым самое большое. Если таких круговых потоков  $C_1$  нет, то добавим ограничение (6) из пункта 2 текущей главы. Если же круговой поток  $C_1$  нашелся, то добавим новую вещественную переменную  $p$  и ограничения (7) – (11);

- Добавим круговой поток  $C$  в список *savedCycles*.

Реализация солвера PathSolver находится в пакете `ru.ifmo.cycles.solver`.

### 2.1.3 Солвер NullSpaceSolver

Солвер NullSpaceSolver развивает идею, на которой базируется солвер PathSolver. В солвере PathSolver мы пытались найти пересечение нового найденного цикла с каким-нибудь из старых циклов. Благодаря этому есть возможность записывать ограничения не о блокировке конкретного цикла, а большого класса циклов. В солвере NullSpaceSolver ищутся не пересечения текущего цикла с одним из старых, а с несколькими старыми, чтобы как можно сильнее уменьшить размер нового ограничения. Алгоритм выглядит следующим образом:

- В найденном векторе потоков  $v$  найдем наибольший круговой поток с помощью солвера CycleSolver, а в нем — наименьший по количеству активных реакций цикл с помощью солвера ContinuousSplitter. Пусть  $d$  — найденный круговой поток.
- Пусть  $c_1, c_2, \dots, c_k$  — найденные ранее круговые потоки. Пусть  $\alpha_1, \alpha_2, \dots, \alpha_k$  — некоторые неотрицательные коэффициенты, и  $d = d' + \sum \alpha_i \cdot c_i$ . Тогда, если все элементы вектора  $d'$  неотрицательны или неположительны, то  $|d'|$  — положительный круговой поток, и мы можем добавить ограничение (6) для потока  $|d'|$ . А иначе, если  $p_1$  и  $p_2$  — множества активных реакций в потоках  $d'_+ = \frac{d'+|d'|}{2}$  и  $d'_- = \frac{|d'|-d'}{2}$  соответственно, то  $d'$  описывает, как вместо пути  $p_2$  в некотором цикле использовать путь  $p_1$ . И тогда мы можем добавить ограничения (19) – (23) для множеств  $A = p_1, B = p_2$ .

- Нужно каким-то образом выбрать коэффициенты  $\alpha_1, \alpha_2, \dots, \alpha_k$  так, чтобы получившиеся ограничения были как можно меньше, то есть, чтобы количество ненулевых элементов вектора  $d' = d - \sum \alpha_i \cdot c_i$  было как можно меньше. Эту задачу решает солвер `LinearSpaceDecomposer`, сведение которого к задаче линейного программирования описано в разделе 2.2.3.

Программная реализация солвера `NullSpaceSolver` представлена классом `NullSpaceSolver` пакета `ru.ifmo.cycles.solver.closed`.

### 2.1.4 Солвер `ClosedSolver`

Солвер `ClosedSolver` базируется на статье [2]. Реализация этого решения находится в классе `OfflineSvdClosedSolver` пакета `ru.ifmo.cycles.solver.closed`. Для нахождения  $N_{int}$  использовалась библиотека `Efficient Java Matrix Library (EJML)` [4].

Также разработана другая реализация солвера, базирующаяся на той же идее. Ее отличие в том, что матрица  $N_{int}$  не находится заранее во время препроцессинга, а находится с помощью `IncumbentCallback` так же, как находятся циклы в солвере `SimpleSolver`. Эта реализация находится в классе `OnlineClosedSolver` пакета `ru.ifmo.cycles.solver.closed`.

## 2.2 Солверы вспомогательных задач линейной оптимизации

В этом разделе описаны солверы задач, возникших по ходу решения основной задачи в солверах, описанных в разделе 2.1.

### 2.2.1 Солвер `CycleSolver`

Задача, решаемая солвером `CycleSolver`, звучит следующим образом: по данному потоку  $v$  требуется найти его положительный круговой подпоток  $v'$ . Множеством реакций в этой задаче является множество внутренних реакций исходной задачи с положительным потоком по ним в данном потоке. Построим следующее сведение:

$$(w, v') \rightarrow \max \quad (13)$$

$$0 \leq v'_i \leq v_i \quad (14)$$



$$S_{int} \cdot v' = 0 \quad (15)$$

$$\exists d > 0: \sum_{i=1}^n v'_i \geq d \quad (16)$$

К сожалению, ограничение (16) не записывается в терминах линейного программирования, так что мы изменим сведение и заодно упростим его, уменьшив количество ограничений. Будем искать не искомый поток  $v'$ , а поток  $v''$ , который отличается от искомого в константу:  $\exists e: v'' = e \cdot v'$ , и при этом сумма его потоков будет хотя бы единица. Заметим, что ограничение (14) теперь не нужно, так как мы можем добиться того, что  $v'_i \leq v_i$ , подбирая достаточно маленькую константу  $e$ . Таким образом, сведение к задаче линейного программирования состоит в следующих трех ограничениях:

$$(w, v'') \rightarrow \max \quad (17)$$

$$S_{int} \cdot v'' = 0 \quad (18)$$

$$\sum_{i=1}^n v''_i \geq 1 \quad (19)$$

Реализацию солвера CycleSolver можно найти в пакете `ru.ifmo.cycles.solver.misc` исходного кода.

### 2.2.2 Солвер ContinuousSplitter

Задача, решаемая солвером ContinuousSplitter, ставится следующим образом: по данному положительному круговому потоку найти наименьшее по количеству элементов подмножество реакций, активных в этом круговом потоке, таких, что существует положительный круговой подпоток, содержащий эти и только эти реакции в качестве активных. Как и в солвере CycleSolver, мы не акцентируем внимание на величине потоков через эти реакции, а лишь на их множестве. Используется следующее сведение к задаче целочисленного линейного программирования:

$$0 \leq v_i \leq M \quad (20)$$

$$a_i \in \{0, 1\} \quad (21)$$

$$a_i \leq v_i \leq M \cdot a_i \quad (22)$$

$$\sum_{i=1}^n a_i \geq 1 \quad (23)$$

$$\sum_{i=1}^n a_i \rightarrow \min \quad (24)$$

В отличие от сведения в солвере SimpleSolver, мы добавили ограничение на то, что если реакция активна с точки зрения переменной  $a_i$ , то величина потока по ней обязана быть ненулевой. Действительно, иначе нулевой поток с одной активной реакцией был бы корректным потоком в рамках этой задачи, но нам такой поток не подходит. Мы ограничиваем величину потока снизу числом 1, если реакция является активной. Мы вправе так сделать опять же потому, что поток можно масштабировать, и любой поток можно умножить на константу так, чтобы величины всех потоков через активные реакции были хотя бы 1, то есть были бы корректным потоком с точки зрения ограничений выше.

Реализацию солвера ContinuousSplitter можно найти в пакете ru.ifmo.cycles.solver.misc исходного кода.

### 2.2.3 Солвер LinearSpaceDecomposer

Солвер LinearSpaceDecomposer решает следующую задачу: по данным векторам  $d, c_1, c_2, \dots, c_k$  длины  $n$  найти набор неотрицательных коэффициентов  $\alpha_1, \dots, \alpha_k$  такой, что вектор  $d' = d - \sum \alpha_i \cdot c_i$  имеет наибольшее возможное количество ненулевых коэффициентов. Эта задача решается следующим сведением к задаче целочисленного линейного программирования. Добавим  $n$  новых целочисленных переменных  $z_1, z_2, \dots, z_n$ . Запишем следующие ограничения:

$$z_i \in \{0, 1\} \quad (25)$$

$$-M^2 \cdot z_i \leq d_i - \sum_{j=1}^k c_{j,i} \leq M^2 \cdot z_i \quad (26)$$

$$\sum z_i \rightarrow \min \quad (27)$$

Очевидно, ограничение (26) говорит о том, что значение  $z_i$  будет равно 1, если  $i$ -я компонента вектора  $d'$  будет отлична от 0. Тогда  $\sum z_i$  будет равна количеству ненулевых компонент  $d'$ , и именно это количество мы хотим минимизировать. Реализацию солвера `LinearSpaceDecomposer` можно найти в пакете `ru.ifmo.cycles.misc` исходного кода.

### 2.3 Выводы по главе 2

В главе 2 были описаны идеи, на которых базируются разработанные солверы, указаны формальные сведения к задаче целочисленного линейного программирования, доказаны нетривиальные переходы между различными сведениями. Были указаны схемы реализации разработанных солверов решаемой задачи, а также побочных задач, возникающих по ходу решения основной задачи.

### ГЛАВА 3. ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

В данной главе произведена проверка правильности разработанных решений, рассмотрены инстансы, на которых тестировались разработанные решения задачи, размеры входных данных и степень разреженности. Также будет проведено сравнение результатов солверов и анализ этих результатов.

#### 3.1 Описание компьютера и программного обеспечения

Все солверы тестировались под операционной системой Windows 7 Service Pack 1 на персональном компьютере Lenovo X230 со следующими характеристиками: Intel® Core™ i5-3320M, 2.60 ГГц, 7.7 Гб ОЗУ.

Все солверы были реализованы на языке программирования Java. Для компиляции и запуска программ использовалась версия Java 1.8.0\_152. Все солверы тестировались, используя один поток выполнения. Реализацию всех солверов можно найти в подпакетах пакета `ru.ifmo.cycles.solver` исходного кода.

#### 3.2 Проверка солверов на маленьком инстансе

Для проверки решений на корректность найденных потоков использовался небольшой инстанс, разработанный специально для этих целей. Для этого инстанса имеется возможность проверить корректность потоков без использования программ, чтобы свести к минимуму вероятность допущения ошибки при реализации солверов.

Инстанс состоит из десяти реакций, две из которых являются внешними, и пяти метаболитов. Инстанс является графовым, то есть каждая реакция, за исключением внешних, в стехиометрической матрице имеет ровно два ненулевых коэффициента: один из них равен 1, а другой  $-1$ . На рисунке 1 показана структура стехиометрической матрицы в виде графа.

Вершинами являются метаболиты, а реакции показаны направленными ребрами, где вершина, из которой идёт ребро, имеет коэффициент  $-1$  в стехиометрической матрице с этой реакцией, а вершина, в которую идет ребро, коэффициент 1. Реакции  $R_9$  и  $R_{10}$  являются внешними.

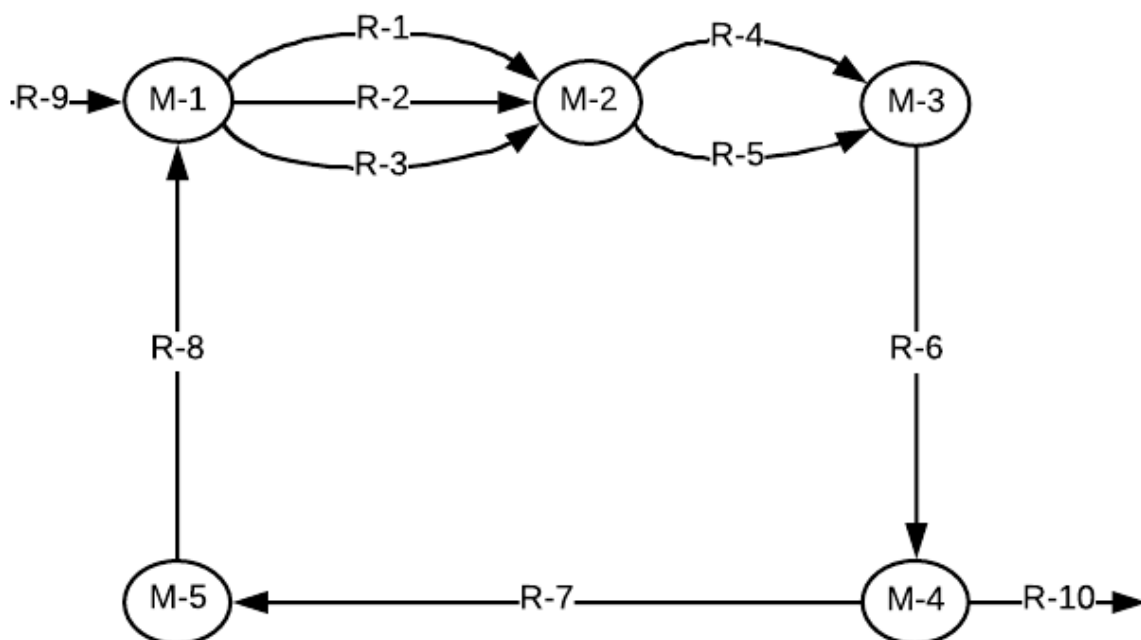


Рисунок 1 — Иллюстрация инстанса

Инстанс имеет следующий вектор весов:  $w = [-1, -1, -1, -1, -1, -1, 1000, -1, 1000, -1]$ . Нижние границы всех реакций равны 0. Верхние границы всех внутренних реакций, то есть реакций  $R_1, R_2, \dots, R_8$ , равны 1000. Верхняя граница реакции  $R_9$  равна 10, реакции  $R_{10}$  — 5. Из рисунка 1 видно, что без учета термодинамических ограничений можно добиться значения целевой функции порядка 1000000. Для этого достаточно пустить поток величины 1000 по циклу  $R_1 - R_4 - R_6 - R_7 - R_8$ . Однако также легко заметить, что если по реакциям  $R_7$  или  $R_8$  идет ненулевой поток, то в данном потоке обязательно будет положительный круговой подпоток. Следовательно, максимальное значение целевой функции имеет порядок около 5000: внешняя реакция  $R_9$  имеет вес 1000, и поток по ней не может быть больше, чем верхняя граница потока через реакцию  $R_{10}$  в силу ограничений баланса. В действительности максимальное значение целевой функции равно 4980, так как на каждую единицу потока по реакции  $R_9$  приходится по единице потока на одну из реакций  $R_1, R_2, R_3$ , по единице потока на одну из реакций  $R_4, R_5$ , по единице потока на реакции  $R_6$  и  $R_{10}$ , а все эти реакции имеют вес  $-1$ .

Округленные до трех знаков после запятой результаты тестирования решения на этом инстансе можно видеть в таблице 1:

Таблица 1 — Результаты работы солверов на инстансе

| Солвер                 | Вектор $v$                                     | Значение целевой функции |
|------------------------|--|--------------------------|
| SimpleSolver           | [0, 0, 5, 0, 5, 5, 0, 0, 5, 5]                 | 4980                     |
| PathSolver             | [0, 0, 5, 0, 5, 5, 0, 0, 5, 5]                 | 4980                     |
| NullSpaceSolver        | [0, 0, 5, 0, 5, 5, 0, 0, 5, 5]                 | 4980                     |
| OfflineSvdClosedSolver | [1.667, 1.667, 1.667, 2.5, 2.5, 5, 0, 0, 5, 5] | 4980                     |
| OnlineClosedSolver     | [0, 5, 0, 5, 0, 5, 0, 0, 5, 5]                 | 4980                     |

Как видно из таблицы, все пять солверов правильно вычислили значение целевой функции. Также из рисунка легко видеть, что все найденные потоки являются корректными потоками в данном инстансе. Действительно, величины потоков по реакциям  $R_7$  и  $R_8$  равны нулю во всех случаях, значит, положительных круговых подпотоков в этих потоках нет. По реакциям  $R_9$ ,  $R_6$  и  $R_{10}$  величины потоков равны пяти. И суммарные величины потоков через множества реакций  $\{R_1, R_2, R_3\}$  и  $\{R_4, R_5\}$  также равны пяти.

### 3.3 Проверка солверов на практических инстансах

#### 3.3.1 Сбор данных об инстансах

Инстансы t1h и max\_297, на которых тестировались решения, были взяты из биоинформатического архива кафедры Компьютерных Технологий Университета ИТМО. Фрагменты этих инстансов представлены на рисунках 2.1 и 2.2. Визуализация инстансов осуществлялась с использованием биоинформатической визуализационной платформы Cytoscape [5].

Множества реакций обоих инстансов соответствуют метаболическим моделям мышинных макрофагов. В t1h вектор весов получен по транскриптомным данным через один час после стимуляции

липополисахаридами. А вектор весов инстанса  $\text{max\_297}$  получен искусственно для максимизации потока по одной реакции  $R_{297}$ .

Каждый инстанс представлен тремя файлами табличного формата:

- `dim.txt` — в этом файле записаны числа  $n$  и  $m$  — размерность задачи;
- `r.txt` — в этом файле для каждой реакции  $i$  записаны числа:  $lb_i$ ,  $rb_i$  и  $w_i$ ;
- `S.txt` — в этом файле записана стехиометрическая матрица.

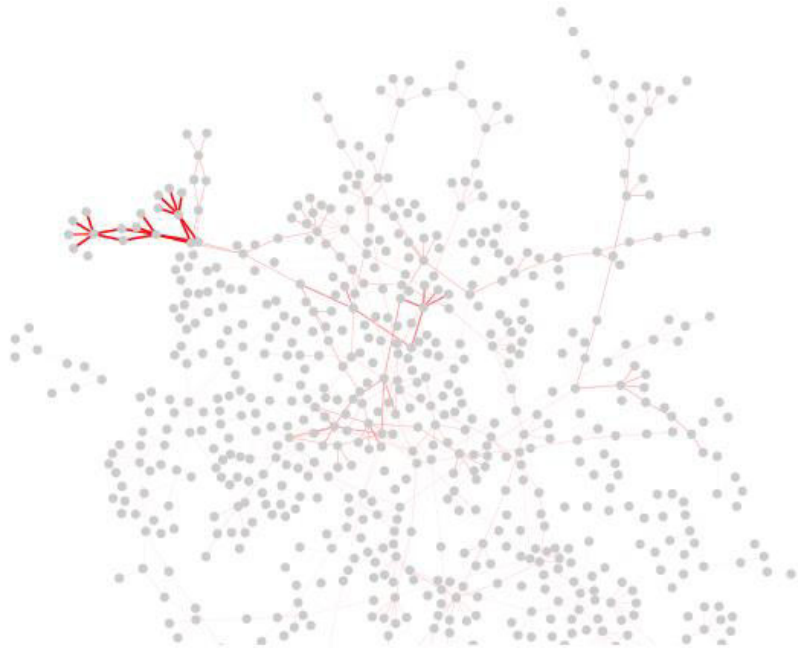


Рисунок 2.1 — Фрагмент инстанса t1h

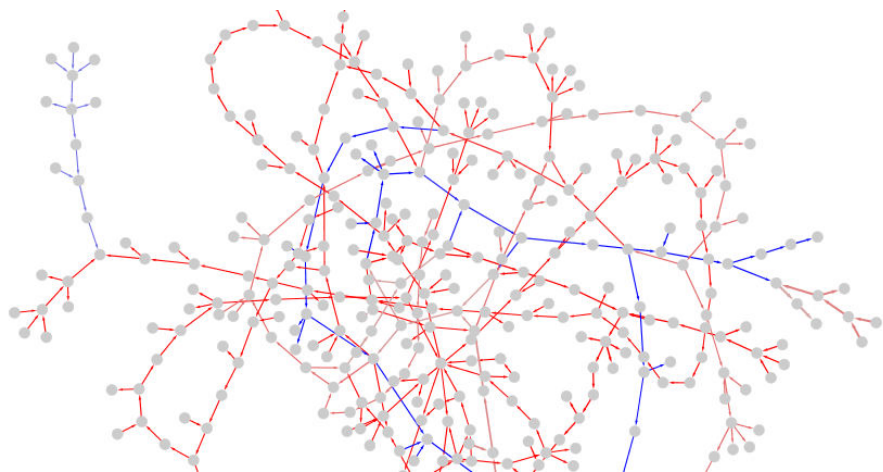


Рисунок 2.2 — Фрагмент инстанса  $\text{max\_297}$

На гистограммах на рисунках 3.1 и 3.2 представлены зависимости количества метаболитов от числа реакций, в которых эти метаболиты

представлены с ненулевым коэффициентом. Данные гистограммы иллюстрируют степень разреженности стехиометрических матриц этих экземпляров. Сбор статистики осуществлялся с помощью библиотеки JFreeChart [6] для языка программирования Java. Исходный код доступен в классе StatCollector пакета ru.ifmo.cycles.misc.

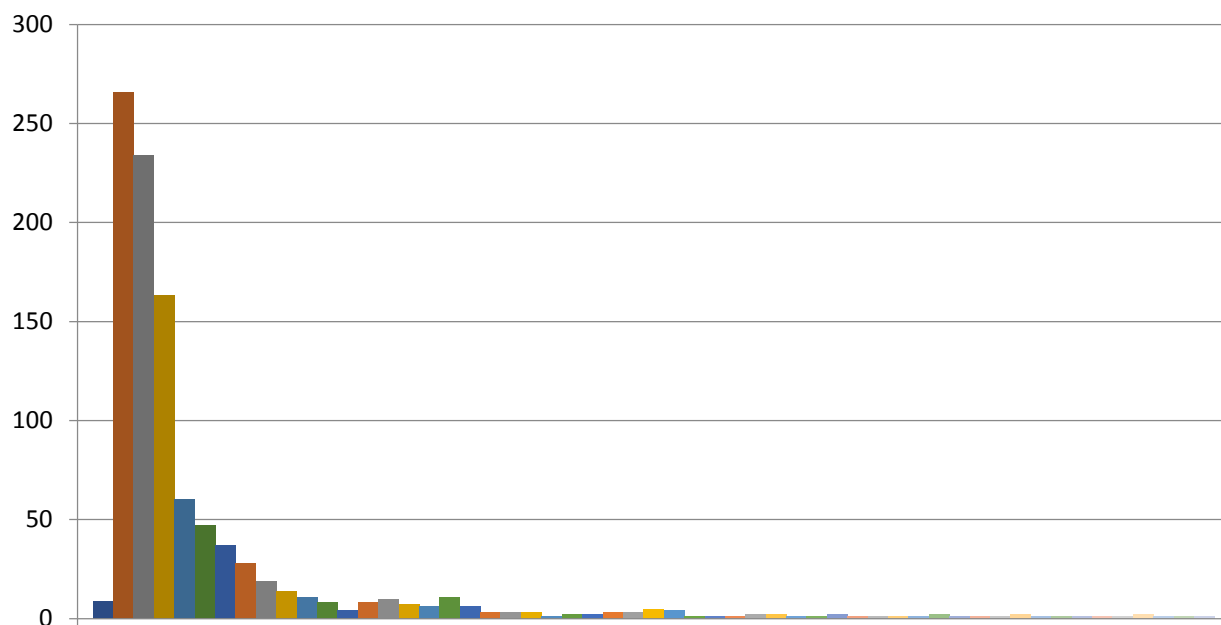


Рисунок 3.1 — Плотность числа реакций в экземпляре t1h

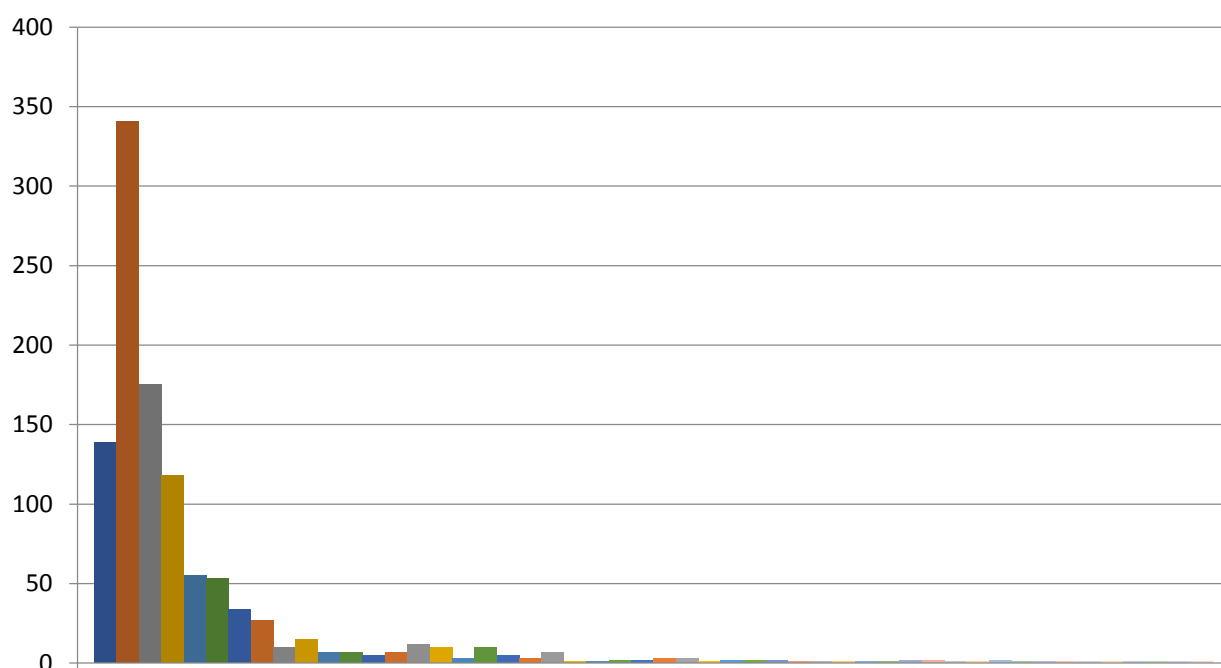


Рисунок 3.2 — Плотность числа реакций в экземпляре max\_297



Как видно из этих гистограмм, большинство метаболитов участвуют в очень ограниченном числе реакций: 95% из них участвуют не более чем в 10 реакциях, что говорит о высокой степени разреженности данных.

Инстанс `t1h` имеет размерность  $1009 \times 2005$ , а инстанс `max_297` имеет размерность  $1084 \times 1808$ . Инстансы имеют примерно одинаковую размерность и примерно одинаковую степень разреженности. Но эти два инстанса имеют отличие: у них очень разная структура вектора весов. У инстанса `t1h` веса «произвольные», их значения похожи на последовательность значений, полученных нормальным распределением, а у инстанса `max_297` вес одной реакции равен 999, а вес остальных равен  $-1$ . Этот инстанс направлен на максимизацию потока через одну фиксированную реакцию  $R_{297}$ , которая нам важна. При этом веса остальных реакций равны  $-1$ , а не 0, потому, что при примерно равных потоках по реакции  $R_{297}$ , предпочтительнее поток, в котором величина потока через другие реакции как можно меньше.

Также для тестирования решений были разработаны инстансы `max_1`, `max_2`, ..., `max_1808`, стехиометрическая матрица которых совпадает с матрицей инстанса `max_297`, а вектор весов инстанса `max_i` максимизирует поток через реакцию  $R_i$ .

### 3.3.2 Тестирование решений на практических инстансах

Для проверки солверов на эффективность каждому солверу было дано два часа на исполнение, после чего он выключался. Если солвер завершался за это время, найденное решение считался результатом его работы. Иначе результатом его работы считалось решение с наибольшим значением целевой функции, найденное солвером за первый час работы.

Результаты работы солверов на инстансе `t1h` можно видеть в таблице 2. Результаты работы солверов на инстансе `max_297` видны в таблице 3. На рисунке 4 представлена диаграмма времени работы солверов SimpleSolver и PathSolver на инстансах `max_1`, `max_2`, ..., `max_1808`. Каждая точка диаграммы представляет один инстанс: по горизонтальной оси отложено время работы

солвера SimpleSolver, по вертикальной оси — солвера PathSolver. Солверы работали 10 минут, после чего выключались.

Таблица 2 — Результаты работы солверов на инстансе t1h

| Солвер                  | Значение целевой функции | Найденная верхняя граница | Время работы, если завершился | Потраченный объем ОЗУ за время работы |
|-------------------------|--------------------------|---------------------------|-------------------------------|---------------------------------------|
| SimpleSolver            | 7.8K                     | 7.8K                      | 16 минут                      | 305 МВ                                |
| PathSolver              | 3.5K                     | 619K                      | Не завершился                 | 1800 МВ                               |
| NullSpaceSolver         | 1.9K                     | 593K                      | Не завершился                 | 1400 МВ                               |
| OnlineClosed-Solver     | 7.3K                     | 1.5M                      | Не завершился                 | 2300 МВ                               |
| OfflineSvd-ClosedSolver | 0                        | 1.8M                      | Не завершился                 | ≈ 0 МВ                                |

Таблица 3 — Результаты работы солверов на инстансе max\_297

| Солвер                  | Значение целевой функции | Найденная верхняя граница | Время работы, если завершился | Потраченный объем ОЗУ за время работы |
|-------------------------|--------------------------|---------------------------|-------------------------------|---------------------------------------|
| SimpleSolver            | 23.1K                    | 995K                      | Не завершился                 | 615 МВ                                |
| PathSolver              | 23.1K                    | 23.1K                     | 52 секунды                    | 2.3 МВ                                |
| NullSpaceSolver         | 15.7K                    | 996K                      | Не завершился                 | 1500 МВ                               |
| OnlineClosed-Solver     | 23.1K                    | 995K                      | Не завершился                 | 473 МВ                                |
| OfflineSvd-ClosedSolver | 0                        | 997K                      | Не завершился                 | ≈0 МВ                                 |

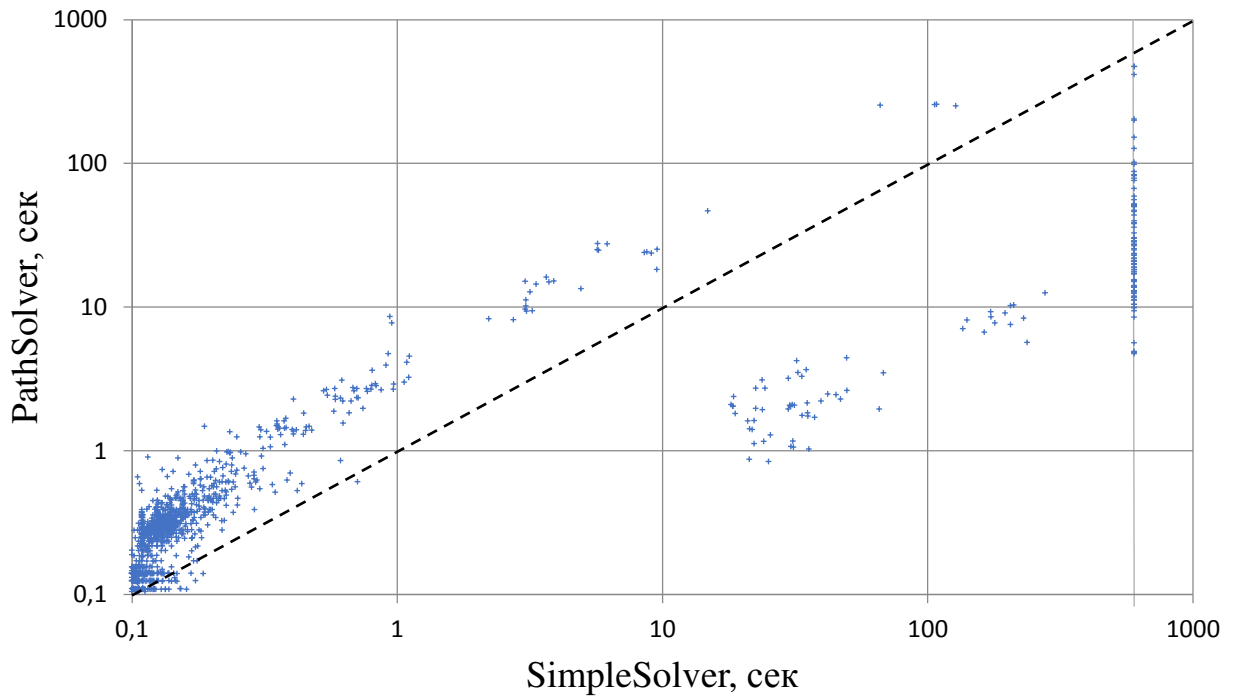


Рисунок 4 — Диаграмма работы солверов SimpleSolver и PathSolver на инстансах  $\max_1, \dots, \max_{1808}$

Как видно из таблицы 2, солвер SimpleSolver — единственный солвер, решивший задачу на инстансе  $t1h$ . При этом среди оставшихся солверов нет ни одного солвера, который смог бы за выделенное время бы найти правильное решение, то есть корректное решение с таким же значением целевой функции, что и решение солвера SimpleSolver.

За дополнительный час исполнения ни один из оставшихся солверов не смог улучшить найденное значение целевой функции или значительно уменьшить верхнюю границу на значение целевой функции. В таблице выделяются результаты работы солвера OfflineSvdClosedSolver. За час работы он не нашел ни одного нетривиального решения задачи. Это связано с тем, что формулировка его сведения к задаче линейного программирования замкнута: все ограничения известны солверу заранее. Этот солвер не ищет промежуточные решения.

Единственный солвер, справившийся с решением задачи на инстансе  $\max_{297}$  за два часа, это солвер PathSolver, причем он справился с задачей меньше чем за одну минуту. Также солверы SimpleSolver и OnlineClosedSolver,

основанные на добавлении ограничений во время работы солвера нашли правильное значение целевой функции, но не смогли доказать, что решения лучше нет. Это связано с тем, что в инстансе `max_297` очень много циклов с похожей структурой, и правила вида «В цикле  $C_1$  можно заменить путь  $P_1$  на путь  $P_2$ » хорошо описывают структуру этих циклов. Впрочем, в инстансе `t1h` тоже много таких циклов. Основное отличие этих инстансов в структуре вектора весов. При решении задачи на инстансе `max_297` метод ветвей и границ предлагает реализованным классам для обратной связи решения, в которых мало активных реакций, так как все реакции, за исключением одной, имеют отрицательный вес в векторе весов  $w$ : добавление каждой активной реакции, кроме одной, ухудшает значение целевой функции. Поэтому найденные циклы имеют очень маленький размер, а значит, и суммы в ограничениях (20) и (22) из главы 2 имеют маленькое число слагаемых.

Солвер `NullSpaceSolver` не смог найти в указанное время корректное решение с наибольшим значением целевой функции ни в одном из инстансов `t1h` и `max_297`, тем более завершиться. Это связано с тем, что в формулировке солвера `LinearSpaceDecomposer` присутствует много ограничений и добавочных переменных, в том числе целочисленных, что негативно сказывается на эффективности работы солвера. Солвер `LinearSpaceDecomposer` используется для нахождения как можно более маленьких ограничений, но размеры найденных ограничений становятся незначительными по сравнению с объемом ресурсов, которые тратит солвер `LinearSpaceDecomposer`. На практике размеры ограничений, найденными солвером `LinearSpaceDecomposer`, не сильно отличаются от размеров ограничений, найденных солвером `PathSolver`, но объем ресурсов для их поиска солвер `PathSolver` намного меньше. Это делает солвер `PathSolver` на несколько порядков более эффективным с точки зрения времени работы и объема памяти, требуемого для полноценной работы солвера.

На инстансах `max_1`, ... `max_1808` все солверы, кроме `PathSolver` и `SimpleSolver` показали себя плохо: эти солверы на большинстве инстансов не

смогли найти решение за 10 минут. Задачу для инстансов, соответствующим точкам, лежащим ниже диагонали, быстрее решил солвер PathSolver, и наоборот. На рисунке 4 видно, что с большинством тестов солверы PathSolver и SimpleSolver справились за время, меньшее одной секунды. На них солвер SimpleSolver был быстрее. Однако большую значимость представляют тесты, на которых солверы работали дольше, так как разница во времени между 5 и 10 минутами более ощутима на практике, чем разница между 1 и 5 секундами. Из рисунка видно, что на таких инстансах солвер PathSolver отработывал быстрее. Также из рисунка видно, что множество на множестве инстансов размера 87 солвер PathSolver справился не более чем за 8 минут, а солвер SimpleSolver не завершился за 10 минут. Из этого эксперимента можно сделать вывод о том, что солвер PathSolver является более универсальным солвером для решения задачи максимизации потока по одной реакции.

### **3.4 Сравнение результатов**

На каждом из тестов был найден разработанный солвер, который значительно превосходит по эффективности существующий солвер OfflineSvdClosedSolver, а также его модернизированную версию OnlineClosedSolver. На произвольных тестах солвер SimpleSolver показывает высокую эффективность, превосходящий на порядок эффективность существующего солвера. На специфическом инстансе, максимизирующем поток по одной реакции, солвер PathSolver показал эффективность, сильно превосходящую эффективность существующего солвера.

### **3.5 Выводы по главе 3**

В главе 3 была описана структура инстансов, на которых тестировались решения. Была осуществлена проверка корректности разработанных солверов на инстансе маленького размера. На больших инстансах разной структуры была проведена проверка и сравнение эффективности различных реализованных солверов: как разработанных в ходе работы, так и солверов, идея которых была взята из статьи [2]. Также было дано обоснование полученных результатов и анализ того, какой солвер эффективнее справляется с каким типом инстансов.

## ВЫВОДЫ

В ходе работы были разработаны солверы SimpleSolver, PathSolver и NullSpaceSolver, а также солверы для задач оптимизации, возникающих в процессе решения основной задачи. Помимо этого было проведено экспериментальное исследование, в ходе которого разработанные солверы сравнивались друг с другом и с существующим солвером.

Основная цель данной работы — разработка эффективного практического солвера — была выполнена. В качестве результата были получены солверы, эффективность которых превосходит эффективность существующего солвера, основанного на результатах статьи [2].

Для достижения цели работы были изучены нюансы работы с пакетом программного обеспечения CPLEX, проведен анализ инстансов, с которыми предстоит иметь дело солверам, опробованы различные подходы к решению задачи.

В дальнейшем работа над данной темой может быть продолжена. Эффективность работы солверов можно повысить, рассматривая другие виды оптимизации метода ветвей и границ: препроцессинг, более эффективные способы обновления нижней границы значения целевой функции.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Страница «Анализ метаболических потоков» [Электронный ресурс] // Интернет-энциклопедия Википедия. 2018. URL: [https://ru.wikipedia.org/wiki/Анализ\\_метаболических\\_потоков](https://ru.wikipedia.org/wiki/Анализ_метаболических_потоков) (дата обращения: 28.05.2018)
2. *Desouki A. A., Jarre F., Gelius-Dietrich G., Lercher M. J.* Cycle free flux: efficient removal of thermodynamically infeasible loops from flux distributions. // *Bioinformatics*. — 2015. — Vol. 31. P. 2159–2165
3. Страница пакета программного обеспечения CPLEX [Электронный ресурс] // Веб-сайт компании IBM. 2018. URL: <https://www.ibm.com/analytics/data-science/prescriptive-analytics/cplex-optimizer> (дата обращения: 28.05.2018)
4. Документация библиотеки Efficient Java Matrix Library для языка программирования Java [Электронный ресурс] // Веб-сайт библиотеки EJML. 2018. URL: <http://ejml.org> (дата обращения: 28.05.2018)
5. Страница биоинформатической визуализационной платформы Cytoscape [Электронный ресурс] // Веб-сайт платформы Cytoscape. 2018. URL: <http://www.cytoscape.org/> (дата обращения: 28.05.2018)
6. Документация библиотеки JFreeChart для языка программирования Java [Электронный ресурс] // Веб-сайт библиотеки JFreeChart. 2018. URL: <http://www.jfree.org/jfreechart/> (дата обращения: 28.05.2018)