

## Пояснительная записка

### Вычислительные техники решения задач линейного программирования в частично-целочисленной постановке и приемы работы с решателем SCIP

Подвойский А.О., Глазунова Е.В.

## Содержание

<b>1</b>	<b>Ключевые термины и определения</b>	<b>3</b>
<b>2</b>	<b>Полезные ресурсы</b>	<b>3</b>
<b>3</b>	<b>Кратко о MPS-формате представления математической постановки задачи</b>	<b>3</b>
3.1	Секция NAME . . . . .	4
3.2	Секция OBJSENSE . . . . .	4
3.3	Секция ROWS . . . . .	4
3.4	Секция COLUMNS . . . . .	4
3.5	Секция RHS . . . . .	5
3.6	Секция BOUNDS . . . . .	5
<b>4</b>	<b>Ключевые компоненты платформы SCIP</b>	<b>5</b>
4.1	Решатель SCIP. Общие сведения . . . . .	5
4.1.1	Установка решателя SCIP . . . . .	5
4.1.2	Приемы работы с решателем SCIP в интерактивной оболочке <code>scip</code> . . . . .	6
4.1.3	Приемы работы с решателем SCIP через обертку <code>PySCIPOpt</code> . . . . .	6
4.2	Декомпозиционный решатель GCG. Общие сведения . . . . .	7
4.2.1	Установка решателя GCG . . . . .	7
4.2.2	Приемы работы с решателем GCG в интерактивной оболочке <code>gcg</code> . . . . .	7
4.2.3	Приемы работы с решателем GCG через обертку <code>PyGCGOpt</code> . . . . .	8
<b>5</b>	<b>Выявленные баги SCIP и тонкости процедуры поиска решения</b>	<b>8</b>
5.1	Недопустимое решение для релаксированной постановки задачи . . . . .	8
5.2	Неединственность релаксированного решения . . . . .	8
5.3	Замечание о стабильности работы решателя SCIP на различных операционных системах . . . . .	8
5.4	Чтение/запись <code>lp</code> -файлов и порядок следования переменных . . . . .	9
<b>6</b>	<b>Альтернативные решатели с открытым исходным кодом</b>	<b>9</b>
6.1	Решатель HIGHS . . . . .	9
6.1.1	Установка решателя на Centos 7 . . . . .	9
6.1.2	Приемы работы с решателем . . . . .	10
6.2	Решатель OPTIMUS (Scala) . . . . .	13

<b>7</b>	<b>Концепт оценки «сложности» задач смешенного целочисленного программирования</b>	<b>13</b>
<b>8</b>	<b>Сжатая сводка результатов вычислительных экспериментов</b>	<b>14</b>
<b>9</b>	<b>Приемы поиска решения</b>	<b>16</b>
9.1	Прием фиксации бинарно-целочисленных переменных в релаксированном решении	16
9.2	Детектирование квазинулевых бинарных и целочисленных переменных . . . . .	17
9.2.1	Предложение по матрице признакового описания объекта . . . . .	17
9.2.2	Анализ важности признаков . . . . .	18
9.3	Прием подавления подгруппы первичных эвристик низкой эффективности . . . . .	20
9.4	Прием подбора порога бинаризации для бинарных переменных в релаксированном решении . . . . .	22
<b>10</b>	<b>Методы машинного обучения в задачах комбинаторной оптимизации</b>	<b>23</b>
10.1	Постановка задачи . . . . .	23
10.2	Концепт матрицы признакового описания бинарных и целочисленных переменных	24
10.3	Стратегии решения задачи . . . . .	24
10.3.1	Стратегия №1. Обнаружение аномалий . . . . .	24
10.3.2	Стратегия №2. Бинарная классификация со слабо выраженным миноритарным классом . . . . .	27
10.4	Трансфер выявленного паттерна. Сценарии группы СОП . . . . .	27
10.4.1	Сценарий tmpfvqodxw.lp без бинарных переменных . . . . .	27
10.4.2	Синтетический сценарий 1664182546_82382.lp с бинарными переменными	28
10.4.3	Синтетический сценарий 1664182533_1587787.lp с бинарными переменными	30
10.4.4	Синтетический сценарий 1664182480_4326847.lp с бинарными переменными	31
10.4.5	Синтетический сценарий 1664182523_380519.lp с бинарными переменными	32
10.5	Концепт построения матрицы признакового описания объекта для задачи выбора стратегии поиска решения . . . . .	33
<b>11</b>	<b>Описание вычислительных экспериментов на сценариях группы ИКП</b>	<b>34</b>
11.1	Поиск решения на сценариях без бинарных переменных.	
	Метаконфигурации SUN, FZBIVSUHPB и ансамбль детекторов аномалий . . . . .	34
11.1.1	Сценарий F398266B без бинарных переменных . . . . .	35
11.1.2	Сценарий 50197DF7 без бинарных переменных . . . . .	36
11.1.3	Сценарий 7FAC4231 без бинарных переменных . . . . .	38
11.1.4	Сценарий CA485A55 без бинарных переменных . . . . .	38
11.1.5	Сценарий 276 без бинарных переменных . . . . .	40
11.1.6	Сценарий 337 без бинарных переменных . . . . .	41
11.1.7	Сценарий 13D686AB без бинарных переменных . . . . .	42
11.1.8	Сценарий A78CBEAD без бинарных переменных . . . . .	44
11.1.9	Сценарий 496 (hard) без бинарных переменных . . . . .	45
11.1.10	Сценарий 514 (hard) без бинарных переменных . . . . .	46
11.1.11	Сценарий 519 (hard) без бинарных переменных . . . . .	47
11.2	Поиск решения на сценариях с бинарными переменными.	
	Метаконфигурация FZBIVSUHPB . . . . .	48

11.2.1	Сценарий A78CBEAD с бинарными переменными . . . . .	50
11.2.2	Сценарий 7FAC4231 с бинарными переменными . . . . .	51
11.2.3	Сценарий 50197DF7 с бинарными переменными . . . . .	52
11.2.4	Сценарий F398266B с бинарными переменными . . . . .	53
11.2.5	Сценарий 337 с бинарными переменными . . . . .	55
11.3	Поиск решения на базе методов машинного и глубокого обучения . . . . .	56
11.3.1	Простое декартово произведение сценариев с бинарными переменными . . .	57
<b>12</b>	<b>Описание вычислительных экспериментов на сценариях группы MBO</b>	<b>59</b>
<b>13</b>	<b>Описание вычислительных экспериментов</b>	
	<b>на сценариях MIPLIB 2017</b>	<b>59</b>
13.1	Сценарии со статусом «open» . . . . .	59
13.1.1	Сценарий DLR2 . . . . .	59
13.1.2	Сценарий CVRPA-N64K9VRPI . . . . .	59
13.2	Сценарии со статусом «hard» . . . . .	59
13.2.1	Сценарий CRYPTANALYSISKB128N50BJ14 . . . . .	59
13.3	Сценарии со статусом «easy» . . . . .	59
13.3.1	Сценарий NEOS-4332801-seret . . . . .	59
	<b>Список иллюстраций</b>	<b>61</b>
	<b>Список таблиц</b>	<b>61</b>
	<b>Список литературы</b>	<b>62</b>

## 1. Ключевые термины и определения

*Сценарий* – это математическая постановка задачи, описанная в терминах математического программирования (например, линейного)

*Сценарий обучающего поднабора* – это сценарий из коллекции сценариев, которые используются на обучающей фазе алгоритма машинного обучения

*Сценарий тестового поднабора* – это сценарий, который используется для построения прогноза с помощью алгоритма машинного обучения

## 2. Полезные ресурсы

<http://plato.asu.edu/sub/benchm.html> Сравнительный анализ эффективности различных решателей от H. D. Mittelmann

## 3. Кратко о MPS-формате представления математической постановки задачи

Больше деталей по ссылке [https://www.gurobi.com/documentation/9.5/refman/mps\\_format.html](https://www.gurobi.com/documentation/9.5/refman/mps_format.html)

MPS-формат – это старейший формат представления математических постановок. Различают два вида этого формата: фиксированный и свободный. В фиксированном формате под имя строки и столбца отводится ровно 8 символов (пробел считается частью имени), а в свободном формате имена могут иметь произвольную длину.

Строки, которые начинаются с символа \* считаются комментариями.

### 3.1. Секция NAME

Первая секция в MPS-файле называется NAME. Она задает имя модели. В фиксированном формате имя модели начинается в 15 столбце.

### 3.2. Секция OBJSENSE

Следующая (необязательная) секция OBJSENSE указывает направление оптимизации (задача решается на минимум или на максимум).

### 3.3. Секция ROWS

В секции ROWS каждая строка описывает *тип ограничения* (E – равенство, L – меньше-или-равно, G – больше-или-равно) и *имя ограничения*. N указывает на целевую функцию.

Самая большая секция MPS-файла это секция COLUMNS, в которой описываются переменные с ненулевыми коэффициентами

```
...
COLUMNS
  z_balance_plus_0_177360  z_balance_0_177360          1  Obj
                           3
  z_balance_plus_100_177446  z_balance_100_177446        1  Obj
                           3
  z_balance_plus_101_177488  Obj                        3
  z_balance_101_177488      1
  z_balance_plus_102_177447  z_balance_102_177447        1  Obj
                           3
...
```

Первая строка говорит, что переменная z\_balance\_plus\_0\_177360 входит в ограничение z\_balance\_0\_177360 с коэффициентом 1, а в целевую функцию – с коэффициентом 3.

### 3.4. Секция COLUMNS

Секция COLUMNS может дополнительно включать маркеры целочисленности. Переменные, расположенные между парой маркеров должны принимать целочисленные значения. Все переменные эти переменные по умолчанию имеют нижнюю границу 0 и верхнюю границу 1. Если требуется задать другие значения нижних и верхних границ, то это можно сделать в секции BOUNDS.

```
INTSTART      'MARKER'                                'INTORG'  *начало секции
  y_tu_170_177684_9_10  out_throughput_vs_units_9          1
  cdo_lower_1_2112221020_1_9_10          1
  y_tu_170_177684_9_10  static_load_terminal_170_177684_9_10  665600
  multiplicity_of_fronts_284          1
  y_tu_170_177684_9_10  cdo_upper_1_2112221020_1_9_10      1  Obj
                           0
  y_tu_170_177685_9_10  cdo_upper_1_2112221020_1_9_10      1
  cdo_lower_1_2112221020_1_9_10          1
...
  y_tu_9_177398_11_9    cdo_lower_1_251194210_1_11_9        1
  multiplicity_of_fronts_345          1
  y_tu_9_177398_11_9    cdo_upper_1_251194210_1_11_9        1
  static_load_terminal_9_177398_11_9    540200
```

```

y_tu_9_177398_11_9      out_throughput_vs_units_8      1  Obj
                        0
INTEND                  'MARKER'                        'INTEND' *конец секции

```

Здесь INTSTART – это имя маркера (игнорируется), 'MARKER' – специальная строка (в одинарных кавычках), 'INTORG' – начало целочисленной секции и 'INTEND' – конце целочисленной секции.

### 3.5. Секция RHS

Следующая секция RHS (right-hand side) MPS-файла описывает правую часть ограничения

```

RHS      cdo_lower_1_2414112504_1_3_3      4
cdo_lower_1_2414112504_1_3_30      4
# cdo_lower_1_2414112504_1_3_3: +1 y_tu_170_177743_3_3 +1 y_tu_170_177744_3_3 >= +4
# cdo_lower_1_2414112504_1_3_30: +1 y_tu_170_177743_3_30 +1 y_tu_170_177744_3_30 >= +4

```

### 3.6. Секция BOUNDS

Это необязательная секция. По умолчанию каждая переменная имеет нижнюю границу 0 и бесконечную верхнюю границу. Каждая строка этой секции может изменять нижнюю границу, верхнюю границу или обе границы переменной.

Поддерживаются следующие типы:

- LO: нижняя граница,
- UP: верхняя граница,
- FX: переменная, зафиксированная в какое-то конкретное значение,
- FR: свободная переменная (без верхней и нижней границы),
- MI: бесконечная нижняя граница,
- PL: бесконечная верхняя граница,
- BV: бинарная переменная (0 или 1),
- LI: нижняя граница для целочисленной переменной,
- UI: верхняя граница для целочисленной переменной,
- SC: верхняя граница для полунепрерывной переменной,
- SI: верхняя граница для полуцелочисленной переменной.

## 4. Ключевые компоненты платформы SCIP

### 4.1. Решатель SCIP. Общие сведения

SCIP (Solving Constraint Integer Programs) <https://www.scipopt.org/> – решатель, предназначенный для решения задач *линейного* и *нелинейного* программирования в частично-целочисленной постановке.

#### 4.1.1. Установка решателя SCIP

Решатель проще всего установить вместе с оберткой PySCIPOpt <https://github.com/scipopt/PySCIPOpt> с помощью менеджеров `pip` или `conda`

```
# установить последнюю доступную версию SCIP
$ pip install pycscipopt
$ conda install -c conda-forge pycscipopt
# установить заданную версию SCIP
$ conda install -c conda-forge pycscipopt=3.4.0
```

#### 4.1.2. Приемы работы с решателем SCIP в интерактивной оболочке scip

Для того чтобы сделать логи более подробными следует включить следующие строки в конфигурационный файл SCIP

scip.set

```
...
display/lpinfo = TRUE
display/ninfeasleaves/active = 2
display/allviols = TRUE
```

#### 4.1.3. Приемы работы с решателем SCIP через обертку PySCIPOpt

Работа над задачей начинается с создания пустого экземпляра модели

```
import pycscipopt

model = pycscipopt.Model()
```

На созданном экземпляре можно вызывать методы чтения модели, конфигурационного файла параметров решателя и т.д.

```
model.readProblem("./problem.lp")
model.readParams("./scip.set")
...
```

С расшифровкой параметров таблицы логов (например, `strbr` или `mdpt`) можно познакомиться на странице проекта GAMS <https://www.gams.com/latest/docs/S SCIP.html>.

Получить коллекцию ограничений в виде списка словарей, содержащих имя переменной и коэффициент при ней в ограничении, можно так

```
model = pycscipopt.Model()
model.readProblem("...")

all_conss: t.Iterable[dict] = [model.getValsLinear(cons) for cons in model.getConss()]
"""
all_conss = [
    { # cons-1
      'cm_or_0_1_1': -1.0,
      'cooked_mass_0_1_1': 1.0,
      'passport_0_1_1': -1.0
    },
    { # cons-2
      'cm_or_0_1_10': -1.0,
      'cm_or_0_1_9': 1.0,
      'cooked_mass_0_1_10': 1.0,
      'passport_0_1_10': -1.0
    },
    ...
]
```

## 4.2. Декомпозиционный решатель GCG. Общие сведения

GCG <https://gcg.or.rwth-aachen.de/#about> – это универсальный декомпозиционный решатель для задач линейного программирования в частично-целочисленной постановке, расширяющий возможности базового решателя SCIP.

Он выявляет структуры в модели, к которым могут быть применены *переформулировка Данцига-Вольфе* или *декомпозиция Бендера*.

Модифицированная постановка задачи (после переформулировки Данцига-Вольфе) решается с помощью обобщения метода ветвей-и-границ, а именно с помощью метода ветвей-штрафов-секущих (branch-price-and-cut), включающего различные механизмы поиска решения – превичные эвристики, стратегии ветвления, стратегии стабилизации, стратегии назначения штрафов и пр.

### 4.2.1. Установка решателя GCG

Проще всего решатель установить вместе с оберткой PyGCGOpt <https://github.com/scipopt/PyGCGOpt> с помощью менеджера пакетов conda

```
$ conda install -c conda-forge pygcgopt
```

### 4.2.2. Приемы работы с решателем GCG в интерактивной оболочке gsg

Прочитать постановку задачи

```
GCG> read problem.lp
```

Запустить процедуру редуцирования размерности

```
GCG> presolve
```

Запустить процедуру поиска структур в матрице ограничений

```
GCG> detect
```

Записать постановку задачи сниженной размерности для gnuplot

```
GCG> write problem problem_reduced.gp
```

Фрагмент gp-файла

```
set encoding utf8
set terminal pdf
set output "problem_reduced.pdf"
set xrange [-1:506441]
set yrange[347788:-1]
set object 1 rect from 0,0 to 506441,183384 fc rgb "#1340C7"
set object 3 rect from 163304,183384 to 163306,183385 fc rgb "#718CDB"
set object 4 rect from 163306,183385 to 163308,183386 fc rgb "#718CDB"
set object 5 rect from 163308,183386 to 163310,183387 fc rgb "#718CDB"
set object 6 rect from 163310,183387 to 163312,183388 fc rgb "#718CDB"
set object 7 rect from 163312,183388 to 163314,183389 fc rgb "#718CDB"
set object 8 rect from 163314,183389 to 163316,183390 fc rgb "#718CDB"
set object 9 rect from 163316,183390 to 163318,183391 fc rgb "#718CDB"
```

```
set object 10 rect from 163318,183391 to 163320,183392 fc rgb "#718CDB"
set object 11 rect from 163320,183392 to 163322,183393 fc rgb "#718CDB"
...
```

Создать pdf-файл декомпозиции задачи после шага снижения размерности

```
$ gnuplot problem_reduced.gp
```

#### 4.2.3. Приемы работы с решателем GCG через обертку PyGCGOpt

## 5. Выявленные баги SCIP и тонкости процедуры поиска решения

### 5.1. Недопустимое решение для релаксированной постановки задачи

По состоянию на 18.06.2022 г. решатель SCIP версии 8.0.0 с оберткой PySCIPOpt версий 4.0.0 и 4.2.0 для операционной системы Windows 10 *релаксированную постановку задачи* (т.е. при снятых ограничениях на целочисленность переменных) оценивает как неспособную привести к допустимому решению.

SCIP версии 7.0.3 (PySCIPOpt 3.4.0) как на операционной системе Windows 10, так и на Unix-подобных операционных системах (в частности, MacOS Monterey 12.1 и Linux Centos 7) решает задачу в релаксированной постановке корректно.

Однако можно попробовать «помочь» SCIP 8.0.X справиться с численными проблемами следующим образом

<https://stackoverflow.com/questions/24702747/scip-infeasibility-detection-with-a-minlp>

scip.set

```
numerics/epsilon = 1e-07 # default 1e-09
numerics/feastol = 1e-05 # default 1e-06
numerics/sumepsilon = 1e-05 # default 1e-06
```

### 5.2. Неединственность релаксированного решения

Если эвристические приемы строятся на базе релаксированного решения задачи, важно помнить, что релаксированные решения, полученные с помощью различных решателей с точки зрения распределения значений переменных могут существенно различаться<sup>1</sup>, не смотря на то, что во всех случаях зазор будет нулевым и целевая функция будет иметь одно и тоже значение (с оговоркой на допуск точности решателя).

### 5.3. Замечание о стабильности работы решателя SCIP на различных операционных системах

- Вычислительные эксперименты проводились на трех версиях решателя SCIP (7.0.0, 7.0.3, 8.0.0) и трех платформах: Windows 10, MacOS (Monterey 12), Linux (Centos 7). Разброс времени поиска решения для каждой конфигурации решателя оценивается минимум по 3 запускам сценария
- На текущий момент наиболее стабильные и наиболее адекватные результаты получаются

<sup>1</sup>Потому как гиперплоскость целевой функции может касаться политопа не в вершине, а по грани



- для ОС Linux (Centos 7) и ОС MacOS (Monterey12) на решателе SCIP версии 7.0.3 (обертка PySCIPOpt 3.4.0) и платформе Ecolle версии 0.7.3 , собранных для однопоточной реализации
- для ОС Windows 10 на решателе SCIP версии 8.0.0 (обертка PySCIPOpt 4.0.0), собранном для однопоточной реализации
- Последняя доступная версия решателя SCIP 8.0.0 (PySCIPOpt 4.1.0) на MacOS (Monterey 12.1) и Linux (Centos 7) при тех же настройках, что и для SCIP версии 7.0.3, как правило, работает значительно медленнее (2.5-2.85 раза) и в большинстве случаев либо не успевает найти решение за отведенное время, либо «просаживает» целевую функцию

## 5.4. Чтение/запись lp-фалов и порядок следования переменных

Чтение/запись файла математической постановки изменяет порядок следования переменных в целевой функции, что может непредсказуемо влиять на эффективность процедуры поиска решения. Если после модификации проблемы с помощью например фиксации нулей сразу запустить процедуру поиска, то будут одни результаты пресолвинга и картина поиска решения, а если модифицированную постановку записать и снова прочитать, то – другие. При этом последствия могут быть очень серьезными: в одном случае решение находится за 20 минут, в другом – за 3 часа. Какой порядок следования переменных «правильный»?

## 6. Альтернативные решатели с открытым исходным кодом

### 6.1. Решатель HIGHS

#### 6.1.1. Установка решателя на Centos 7

Установить решатель HIGHS <https://ergo-code.github.io/HiGHS/get-started.html> можно следующим образом

1. Скачать репозиторий проекта

```
$ git clone https://github.com/ERGO-Code/HiGHS.git
```

2. Установить cmake версии >=3.15

```
# https://cmake.org/download/
$ wget https://github.com/Kitware/CMake/releases/download/v3.24.2/cmake-3.24.2.tar.gz
$ tar -xvf cmake-3.24.2.tar.gz
$ cd cmake-3.24.2
$ ./bootstrap --prefix=/usr --datadir=share/cmake --docdir=doc/cmake && make
$ sudo make install
$ cmake --version # cmake version 3.24.2
```

3. Установить альтернативную версию компилятора gcc (например, версии 7) для сборки проекта

```
# https://linuxize.com/post/how-to-install-gcc-compiler-on-centos-7/
$ gcc --version # gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
```

Чтобы получить доступ к альтернативной версии компилятора GCC 7, требуется запустить новый сеанс командной оболочки с помощью утилиты `scl`

```
$ scl enable devtoolset-7 bash
# или для ZSH
# scl enable devtoolset-7 zsh
$ gcc --version # gcc (GCC) 7.3.1 20180303 (Red Hat 7.3.1-5)
```

- В директории проекта HiGHS создать поддиректорию `build` и запустить из-под нее утилиту `cmake`

```
$ cd HiGHS
$ mkdir build
$ cd build
$ cmake -DFAST_BUILD=ON ..
$ cmake --build .
# Чтобы убедиться в том, что сборка прошла успешно, рекомендуется запустить быстрые тесты
$ ctest
```

В результате будет создан исполняемый файл `build/bin/highs`

- Добавить путь до утилиты в конфигурационный файл оболочки

`.zshrc`

```
...
export PATH=${HOME}/Projects/HiGHS/build/bin:$PATH
```

После внесения изменений в конфигурационный файл, можно пересчитать конфигурацию сессии

```
$ source .zshrc
```

### 6.1.2. Приемы работы с решателем

Для запуска решателя в MILP-режиме требуется только передать путь до `*.lp/*.mps`-файла

```
$ highs /path/to/model.lp
$ highs --help
```

Для запуска решателя в режиме поиска релаксированного решения требуется параметру `--solver` передать название метода (на текущий момент поддерживается только симплекс-метод `simplex` и метод внутренней точки `ipm`)

```
# LP-задача будет решаться методом внутренней точки
$ highs --solver ipm --model_file 50197df7_bin.lp
```

Запуск решателя в параллельном MILP-режиме, с шагом снижения размерности задачи и ограничением по времени расчета будет выглядеть так

```
$ highs \
  --model_file 514.lp \
  --presolve on \
  --parallel on \
  --time_limit 950 \
  --solution_file highs_514.sol
```

Список управляющих параметров решателя доступен на странице документации HiGHS для интерфейса Rust <https://www.maths.ed.ac.uk/hall/HiGHS/HighsOptions.html>

## HiGHS Options

### presolve

Presolve option: "off", "choose" or "on"

type: string, advanced: false, default: "choose"

### solver

Solver option: "simplex", "choose" or "ipm"

type: string, advanced: false, default: "choose"

### parallel

Parallel option: "off", "choose" or "on"

type: string, advanced: false, default: "choose"

### time\_limit

Time limit

type: double, advanced: false, range: [0, inf], default: inf

### ranging

Compute cost, bound, RHS and basic solution ranging: "off" or "on"

type: string, advanced: false, default: "off"

### infinite\_cost

Limit on cost coefficient: values larger than this will be treated as infinite

type: double, advanced: false, range: [1e+15, inf], default: 1e+20

### infinite\_bound

Limit on |constraint bound|: values larger than this will be treated as infinite

type: double, advanced: false, range: [1e+15, inf], default: 1e+20

### small\_matrix\_value

Lower limit on |matrix entries|: values smaller than this will be treated as zero

type: double, advanced: false, range: [1e-12, inf], default: 1e-09

### large\_matrix\_value

Upper limit on |matrix entries|: values larger than this will be treated as infinite

type: double, advanced: false, range: [1, inf], default: 1e+15

### primal\_feasibility\_tolerance

Primal feasibility tolerance

type: double, advanced: false, range: [1e-10, inf], default: 1e-07

### dual\_feasibility\_tolerance

Dual feasibility tolerance

type: double, advanced: false, range: [1e-10, inf], default: 1e-07

### ipm\_optimality\_tolerance

IPM optimality tolerance

type: double, advanced: false, range: [1e-12, inf], default: 1e-08

### objective\_bound

Objective bound for termination

type: double, advanced: false, range: [-inf, inf], default: inf

### objective\_target

Objective target for termination

type: double, advanced: false, range: [-inf, inf], default: -inf

### random\_seed

random seed used in HiGHS

type: HighsInt, advanced: false, range: {0, 2147483647}, default: 0

### threads

number of threads used by HiGHS (0: automatic)

type: HighsInt, advanced: false, range: {0, 2147483647}, default: 0

### highs\_debug\_level

Debugging level in HiGHS

type: HighsInt, advanced: false, range: {0, 3}, default: 0

### highs\_analysis\_level

Analysis level in HiGHS

type: HighsInt, advanced: false, range: {0, 63}, default: 0

### simplex\_strategy

Strategy for simplex solver

type: HighsInt, advanced: false, range: {0, 4}, default: 1

### simplex\_scale\_strategy

Simplex scaling strategy: off / choose / equilibration / forced equilibration / max value 0 / max value 1 (0/1/2/3/4/5)  
type: HighsInt, advanced: false, range: {0, 5}, default: 1  
simplex\_crash\_strategy  
Strategy for simplex crash: off / LTSSF / Bixby (0/1/2)  
type: HighsInt, advanced: false, range: {0, 9}, default: 0  
simplex\_dual\_edge\_weight\_strategy  
Strategy for simplex dual edge weights: Choose / Dantzig / Devex / Steepest Edge (-1/0/1/2)  
type: HighsInt, advanced: false, range: {-1, 3}, default: -1  
simplex\_primal\_edge\_weight\_strategy  
Strategy for simplex primal edge weights: Choose / Dantzig / Devex (-1/0/1)  
type: HighsInt, advanced: false, range: {-1, 1}, default: -1  
simplex\_iteration\_limit  
Iteration limit for simplex solver  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 2147483647  
simplex\_update\_limit  
Limit on the number of simplex UPDATE operations  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 5000  
ipm\_iteration\_limit  
Iteration limit for IPM solver  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 2147483647  
simplex\_min\_concurrency  
Minimum level of concurrency in parallel simplex  
type: HighsInt, advanced: false, range: {1, 8}, default: 1  
simplex\_max\_concurrency  
Maximum level of concurrency in parallel simplex  
type: HighsInt, advanced: false, range: {1, 8}, default: 8  
output\_flag  
Enables or disables solver output  
type: bool, advanced: false, range: {false, true}, default: true  
log\_to\_console  
Enables or disables console logging  
type: bool, advanced: false, range: {false, true}, default: true  
solution\_file  
Solution file  
type: string, advanced: false, default: ""  
log\_file  
Log file  
type: string, advanced: false, default: "Highs.log"  
write\_solution\_to\_file  
Write the primal and dual solution to a file  
type: bool, advanced: false, range: {false, true}, default: false  
write\_solution\_style  
Write the solution in style: 0=>Raw (computer-readable); 1=>Pretty (human-readable)  
type: HighsInt, advanced: false, range: {0, 1}, default: 0  
mip\_detect\_symmetry  
Whether symmetry should be detected  
type: bool, advanced: false, range: {false, true}, default: true  
mip\_max\_nodes  
MIP solver max number of nodes  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 2147483647  
mip\_max\_stall\_nodes  
MIP solver max number of nodes where estimate is above cutoff bound  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 2147483647  
mip\_max\_leaves  
MIP solver max number of leave nodes  
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 2147483647  
mip\_lp\_age\_limit  
maximal age of dynamic LP rows before they are removed from the LP relaxation  
type: HighsInt, advanced: false, range: {0, 32767}, default: 10

```

mip_pool_age_limit
maximal age of rows in the cutpool before they are deleted
type: HighsInt, advanced: false, range: {0, 1000}, default: 30
mip_pool_soft_limit
soft limit on the number of rows in the cutpool for dynamic age adjustment
type: HighsInt, advanced: false, range: {1, 2147483647}, default: 10000
mip_pscost_minreliable
minimal number of observations before pseudo costs are considered reliable
type: HighsInt, advanced: false, range: {0, 2147483647}, default: 8
mip_report_level
MIP solver reporting level
type: HighsInt, advanced: false, range: {0, 2}, default: 1
mip_feasibility_tolerance
MIP feasibility tolerance
type: double, advanced: false, range: [1e-10, inf], default: 1e-06
mip_heuristic_effort
effort spent for MIP heuristics
type: double, advanced: false, range: [0, 1], default: 0.05

```

TODO: При запуске решателя в режиме поиска релаксированного решения процесс зависает, если заданы параметры, управляющие подробностью вывода

- `highs_debug_level = 2`,
- `mip_report_level = 2`

Задать значения управляющих параметров можно в set-файле HiGHS

highs.set

```

time_limit = 7200
simplex_iteration_limit = 10000
ipm_iteration_limit = 5000
...

```

Теперь для запуска решателя в специфицированном режиме остается только передать путь до файла настроек параметру `--options_file`

```
$ highs --model_file 496.lpl --options_file highs.set
```

## 6.2. Решатель OPTIMUS (Scala)

## 7. Концепт оценки «сложности» задач смешенного целочисленного программирования

Постановка: требуется разработать концепт вычисления метрики, оценивающей трудность решения задачи смешенного целочисленного программирования.

Найденное значение метрики можно использовать для привязки вычислительных техник поиска решения к семейству проблем и корректировке процедуры поиска, а также для оценки качества фиксации.

Представляется разумным о метрике рассуждать в терминах матрицы ограничений и статистик, полученных в ходе пресолвинга и решения исходной задачи в релаксированной постановке, не привязанных к бизнес-интерпретациям.

В ходе процедуры поиска решения значения верхней *UB* и нижней *LB* границ изменяются во времени. Может оказаться, что первое «быстро» найденное допустимое решение с зазором

близким к требованиям проекта не удастся развить в течение долгого времени. Или наоборот, первое допустимое решение находится за несколько минут до исчерпания бюджета по времени, но с зазором более близким к оптимальному, чем указано в требованиях проекта.

Таким образом, чтобы учесть и временную компоненту, и зазор найденного решения кажется разумным учитывать площадь под кривой изменения зазора во времени с учетом поправки на время поиска первого допустимого решения, и называть полученную величину *интегральной оценкой сложности поиска решения по версии решателя X* или *фактическим барьером задачи по версии решателя X*

$$B_t := G^{1-\tau_0/\tau'} \in (0, 1],$$

где  $\tau_0$  – время поиска первого допустимого решения,  $\tau'$  – бюджет по времени ( $\tau' = 7200$ , секунд),  $G$  – площадь под кривой изменения зазора во времени.

Кривая строится в плоскости безразмерных параметров « $\tau_0/\tau' - g/g'$ », где  $g$  – текущее значение зазора в момент времени  $\tau$ , а  $g'$  – пороговое значение зазора ( $g' = 100\%$ ).

Таким образом, если первое допустимое решение удастся найти в самом начале процедуры поиска ( $\tau_0 \approx 0$ ), то остается только следить за площадью под кривой изменения зазора во времени  $G$ . А если первое допустимое решение находится ближе к моменту исчерпания бюджета по времени, то  $1 - \tau_0/\tau' \approx 0$ , и следовательно энергия поиска решения будет близка к своему максимальному значению  $B_t \approx 1$ .

Грубое приближение оценки барьера задачи  $B_p$  в аналитической форме предлагается вычислять следующим образом

$$B_p := \left(\frac{\tau_{\text{relax}}}{\tau'}\right)^{1/3} \cdot \exp\left(1 - \frac{\#zero\_ints_{\text{relax}}}{\#ints_{\text{after}}}\right) \cdot \frac{\#conss_{\text{after}}}{\#conss_{\text{before}}} \cdot \exp\left(1 - \frac{\gamma_{\text{before}}}{\gamma_{\text{after}}}\right)^3,$$

где

$\tau_{\text{relax}}$  – временные издержки на поиск релаксированного решения,

$\tau_{\text{presolve}}$  – временные издержки на пресолвинг,

$\#ints_{\text{before}}$  – количество целочисленных переменных до пресолвинга,

$\#zero\_ints_{\text{relax}}$  – количество нулевых целочисленных переменных в релаксированном решении,

$\#conss_{\text{before}}$  – количество ограничений до пресолвинга,

$\#conss_{\text{after}}$  – количество ограничений после пресолвинга,

$\gamma_{\text{before}}$  – плотность ненулевых коэффициентов в матрице ограничений до пресолвинга,

$\gamma_{\text{after}}$  – плотность ненулевых коэффициентов в матрице ограничений после пресолвинга.

## 8. Сжатая сводка результатов вычислительных экспериментов

Все эксперименты проводились на ОС Linux Centos 7 Intel Core™ i7 (8 CPUs), 3.6GHz; RAM 32Gb. Использовался MILP-решатель SCIP 7.0.3 (Python-обертка PySCIPOpt 3.4.0) и Python 3.8.0.

Развернутая сводка результатов приводится по ссылке [https://docs.google.com/document/d/16p8\\_VjZaHCBdDwo\\_YNZaEpZVFgmLyDi5A6104gX3zK8/edit?usp=sharing](https://docs.google.com/document/d/16p8_VjZaHCBdDwo_YNZaEpZVFgmLyDi5A6104gX3zK8/edit?usp=sharing)

Обозначения

- CBC+DOH – доменно-ориентированные эвристики, работающие поверх решателя CBC.
- CBC+MS – мера подобия релаксированного решения, работающая поверх решателя CBC.

- SCIP(d) – решатель SCIP с настройками по умолчанию.
- SUN – метаконфигурация, работающая поверх решателя SCIP: подавляется подгруппа первичных эвристик низкой эффективности.
- FZBIVSUNPB – метаконфигурация, работающая поверх решателя SCIP: подавляется подгруппа первичных эвристик низкой эффективности; при ветвлении и разрешении конфликтов решатель отдает предпочтение бинарными переменным; фиксируются нулевые бинарные и целочисленные переменные релаксированного решения.
- EAD(contamination; file\_name) – модель машинного обучения, работающая поверх решателя SCIP: подавляется подгруппа первичных эвристик низкой эффективности; при ветвлении и разрешении конфликтов решатель отдает предпочтение бинарными переменным; частично-заданное решение на фиксациях строится на основании прогноза ансамбля детекторов аномалий; contamination – доля аномальных экземпляров в наборе данных, file\_name – имя lp-файла математической постановки, на котором обучался ансамбль детекторов аномалий.
- Detector\_name(contamination; file\_name) – детектор аномалий, работающий поверх решателя SCIP: подавляется подгруппа первичных эвристик низкой эффективности; при ветвлении и разрешении конфликтов решатель отдает предпочтение бинарными переменным; частично-заданное решение на фиксациях строится на основании прогноза детектора аномалий; contamination – доля аномальных экземпляров в наборе данных, file\_name – имя lp-файла математической постановки, на котором обучался детектор аномалий.
- RELAX - релаксированное решение, найденное с помощью решателя SCIP.

## Выводы

1. На всех сценариях группы ИКП метаконфигурация FZBIVSUNPB помогает решателю SCIP найти более низкое значение целевой функции и за меньшее время.
2. На всех сценариях группы ИКП (за исключением сценариев a78cbead\_bin.lp, 7fac4231\_bin.lp и 50197df7\_bin.lp) ансамбль детекторов аномалий без подбора параметра контаминации EAD(0.10; f398266b\_bin.lp) помогает решателю SCIP найти более низкое значение целевой функции и за меньшее время. На сценариях a78cbead\_bin.lp, 7fac4231\_bin.lp и 50197df7\_bin.lp прием EAD не смог найти решение за отведенное время.
3. На сценариях 514.lp, 519.lp, a78cbead\_bin.lp, 7fac4231\_bin.lp и 50197df7\_bin.lp изолированные детекторы аномалий помогают решателю SCIP найти более низкое значение целевой функции и за меньшее время.
4. На сценариях 514.lp, 519.lp, a78cbead\_bin.lp, 7fac4231\_bin.lp и 50197df7\_bin.lp изолированные детекторы аномалий находят решения, которые по сравнению с решениями, полученными средствами CBC+DOH(MS), оказываются лучше в среднем на 50.73% по временным издержкам и в среднем на 6.32% – по целевой функции.
5. На всех сценариях группы ИКП (за исключением сценариев 514.lp и 519.lp) метаконфигурация FZBIVSUNPB находит решения, которые оказываются нехуже решений, полученных с помощью CBC+DOH(MS), как с точки зрения полного времени расчета (среднее улучшение 62.16%), так и с точки зрения целевой функции (среднее улучшение 7.03%). На сценарии 514.lp метаконфигурация получает решение, которое только по целевой функции (+18.616%) превосходит решение, найденное средствами CBC+DOH(MS). На сценарии 519.lp решение метаконфигурации уступает решению, найденному с помощью CBC+DOH(MS) и по временным издержкам (-14.29%) и по целевой функции (-2.302%).



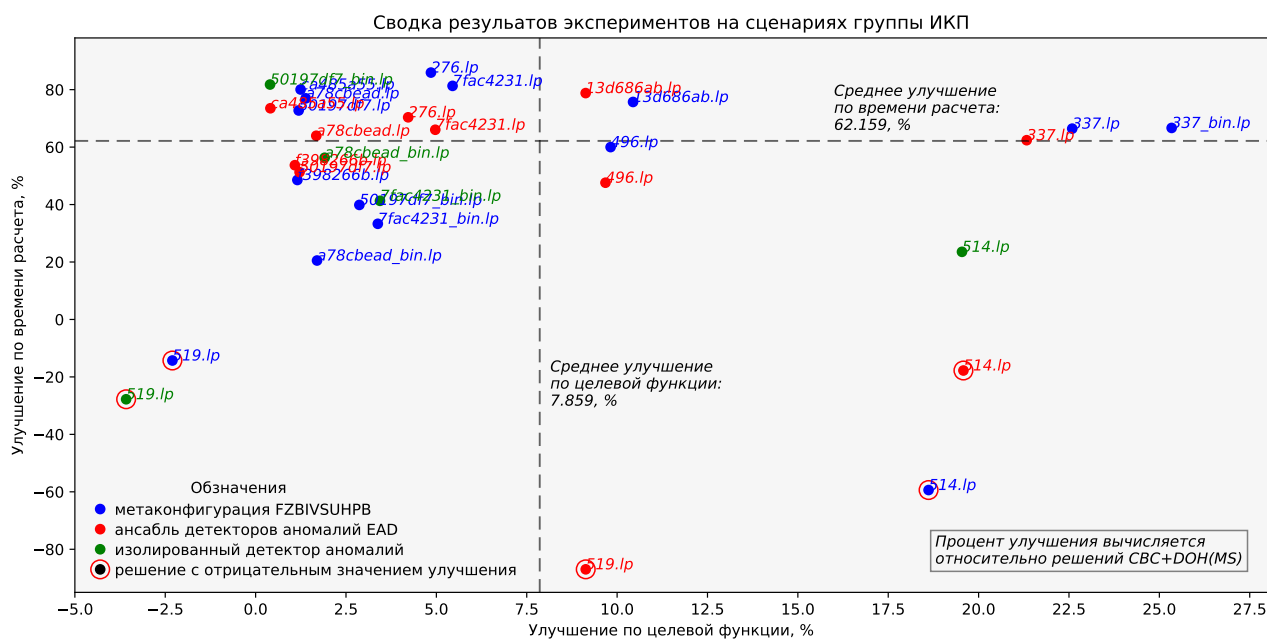


Рис. 1. Сводка результатов вычислительных экспериментов на сценариях группы ИКП

## 9. Приемы поиска решения

### 9.1. Прием фиксации бинарно-целочисленных переменных в релаксированном решении

Часто фиксация целочисленных переменных<sup>2</sup> в релаксированном решении приводит к приемлемому допустимому целочисленному решению, которое потом можно использовать как «теплый старат» или как базовое решение для других схем фиксации.

```

ZERO = 0.0
...
relax_sol: pd.Series = read_relax_sol(path_to_relax_sol)

model = pyscipopt.Model()
model.readProblem(path_to_lp_file)
model.readParams(path_to_set_file)

all_vars: t.List[pyscipopt.scip.Variable] = model.getVars()
bin_vars: t.List[pyscipopt.scip.Variable] = extract_vars_set_type(all_vars, BINARY)
int_vars: t.List[pyscipopt.scip.Variable] = extract_vars_set_type(all_vars, INTEGER)

all_zero_bin_vars: t.List[
    pyscipopt.scip.Variable
] = extract_from_relax_sol_zero_vars(
    relax_sol,
    sub_group_vars=bin_vars,
)
all_zero_int_vars: t.List[
    pyscipopt.scip.Variable
] = extract_from_relax_sol_zero_vars(
    relax_sol,
    sub_group_vars=int_vars,
)

```

<sup>2</sup>Вообще говоря, фиксировать можно не только бинарные и целочисленные переменные



```
for var in all_zero_bin_vars + all_zero_int_vars:
    model.fixVar(var, ZERO)

model.optimize()
...
```

## 9.2. Детектирование квазинулевых бинарных и целочисленных переменных

### 9.2.1. Предложение по матрице признакового описания объекта

Проблема «квазинулевых» переменных заключается в том, что некоторые бинарные и целочисленные переменные, которые принимают нулевое значение в релаксированном решении, могут принять ненулевые значения в целочисленном решении (такие переменные предлагается считать *аномалиями*). Фиксация, построенная на таких «ложных нулях» с одной стороны обладает высокой компрессией как по переменным, так и по ограничениям (то есть значительно снижает размерность задачи), но с другой – сильно мешает решателю.

Задача сводится к тому, чтобы научиться отличать одну нулевую бинарную или целочисленную переменную от другой. Кажется естественным для этой задачи использовать аппарат машинного обучения. Строки матрицы плана будут описывать контекст переменных и, как ожидается, помогут детектировать квазинулевые переменные с высокой полнотой и точностью.

Матрицу признакового описания объекта можно строить на переменных, которые приняли нулевое значение в релаксированном решении, найденном с помощью SCIP, и ненулевое значение в целочисленном решении, найденном с помощью коммерческого решателя (CPLEX, Gurobi, CORT etc.). Такие переменные будем называть аномалиями и помечать единицей. Для построения задачи бинарной классификации на сбалансированном наборе данных остается только из исходной матрицы плана выбрать «штатные» экземпляры (помечаются 0) в количестве равном количеству аномалий.

Предлагается использовать следующие признаки:

1. `relax_sol` (то же, что и `var_lp_sol_relax`): значения переменных в релаксированном решении,
2. `var_ub_global_relax`: глобальные верхние границы переменных в релаксированном решении,
3. `var_lp_sol_default_presolve`: значения переменных в релаксированном решении, полученном после пресолвинга с настройками по умолчанию,
4. `n_vars_after_presolving_default_presolve`: число переменных после пресолвинга с настройками по умолчанию.
5. `continuous_context_types`: число вещественных переменных, входящих в те же ограничения, что и рассматриваемая переменная,
6. `binary_context_types`: число бинарных переменных, входящих в те же ограничения, что и рассматриваемая переменная,
7. `integer_context_types`: число целочисленных переменных, входящих в те же ограничения, что и рассматриваемая переменная,
8. `var_mean_relax_context`: значения релаксированного решения, усредненные по переменным, входящим в те же ограничения, что и рассматриваемая переменная.

### 9.2.2. Анализ важности признаков

Лучший набор признаков, упорядоченный по убыванию *пермутационной важности*, найденной с помощью модели случайного леса (`n_estimators=450`, `max_features=0.25`, `max_depth=8`)<sup>3</sup>:

1. `integer_context_after_presolving`,
2. `var_ub_global_after_presolving`,
3. `relax_sol`,
4. `var_ub_global_relax`,
5. `var_context_mean_relax_after_presolving`.

Важность признаков *по Шепли*, найденная с помощью Zero-shot AutoML LightGBM библиотеки FLAML (`boosting_type="gbdt"`, `objective="binary"`, `max_depth=7`) , приведена на рис. 2.

```
clf_zero_shot = default.LGBMClassifier(  
    boosting_type="gbdt",  
    objective="binary",  
    max_depth=7, # приходится помочь; в противном случае сильно переобучается  
)  
  
clf_zero_shot.fit(X_sub_train, y_sub_train)  
  
explainer = shap.TreeExplainer(clf_zero_shot)  
shap_values_train = explainer.shap_values(X_sub_train)  
shap_values_val = explainer.shap_values(X_val)  
  
shap.summary_plot(shap_values_train[1], X_sub_train)
```

Как показывает глобальная интерпретация (рис. 2) большие значения верхней границы после пресолвинга (признак `var_ub_global_after_presolving`) отвечают «аномальным» переменным. То есть другими словами чем выше верхняя граница нулевой бинарной или целочисленной переменной, тем вероятнее она примет ненулевое значение в целочисленном решении. Напротив, меньшие значения признака `integer_context_after_presolving` указывают на «аномальность» переменной – в среднем чем меньше целочисленных переменных в контексте рассматриваемой переменной, тем выше вероятность, что она окажется аномальной.

Ниже приводятся графики частичной зависимости признаков. График частичной зависимости признака `relax_sol` (рис. 5) показывает, что в целом низкие значения переменной в релаксированном решении отвечают «штатным» переменным, но есть небольшой кластер точек, которые даже если и принимают нулевое или близкое к нулю значение в релаксированном решении, все-таки оказываются аномальными. Это и есть искомые «квази нулевые» переменные. Здесь есть неоднозначность. Среди «штатных» бинарных и целочисленных переменных чаще оказываются такие, в контексте которых мало целочисленных переменных, но есть и такие в контексте которых целочисленные переменные встречаются часто.

```
shap.dependence_plot("feature_name", shap_values_val[1], X_val)
```

Пострим локальную интерпретацию на экземплярах. Для проверки вычислим значения Шепли с помощью встроенных средств LightGBM и с помощью библиотеки SHAP. Важно помнить, что в отличие от библиотеки SHAP, метод `predict` с флагом `predict_contib=True` библиотеки LightGBM возвращает матрицу размером  $n\_samples \times (n\_features + 1)$ , а не  $n\_samples \times n\_features$ .

<sup>3</sup>На обучающем поднаборе: гармоническое среднее: 0.687, полнота: 0.6370, точность: 0.745, площадь под ROC-кривой: 0.709, на валидационном поднаборе: гармоническое среднее: 0.670, полнота: 0.628, точность: 0.720, площадь под ROC-кривой: 0.692

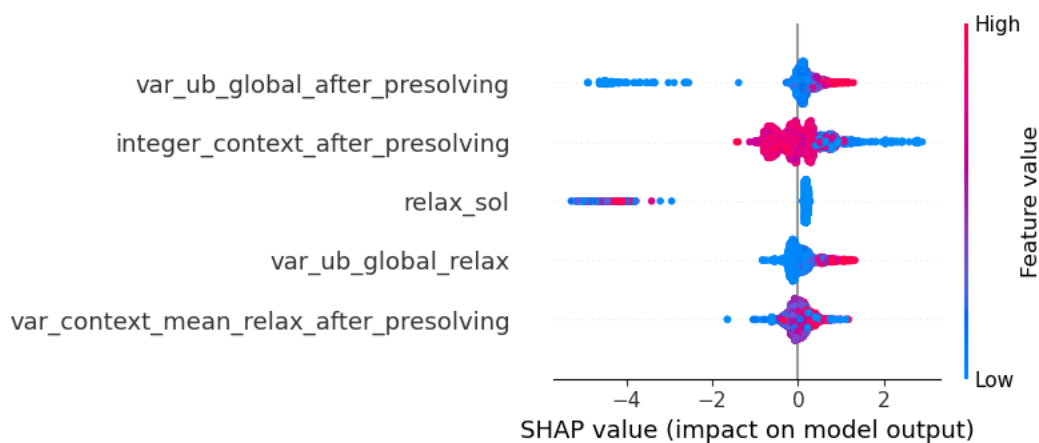


Рис. 2. Важность признаков по Шепли

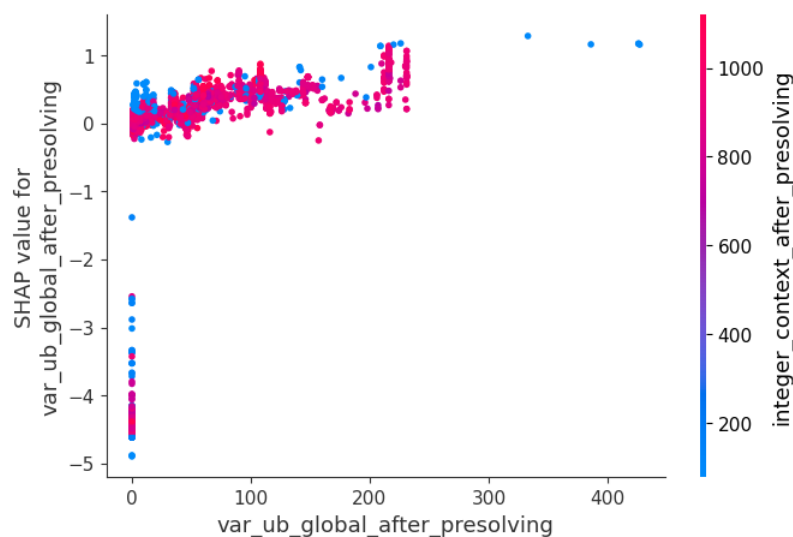


Рис. 3. График частичной зависимости признака var\_ub\_global\_after\_presolving от признака integer\_context\_after\_presolving

Последний столбец в матрице это ожидаемое значение (expected value<sup>4</sup>) <https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html#lightgbm.LGBMClassifier>.

```
clf_zero_shot.predict(
    X_val,
    pred_contrib=True
)[: , :-1].sum(axis=1)[SAMPLE_IDX] # -0.49065584326131984

shap_values_val[1][SAMPLE_IDX, :].sum() # -0.49065584326131984
```

И, наконец, строим локальную интерпретацию для первого экземпляра валидационного поднабора (см. рис. 6)

```
SAMPLE_IDX = 0

shap.force_plot(
    explainer.expected_value[1],
    shap_values_val[1][SAMPLE_IDX, :],
    X_val.iloc[SAMPLE_IDX, :]
)
```

<sup>4</sup> Оно же на графиках локальной интерпретации библиотеки SHAP обозначается как «base value»

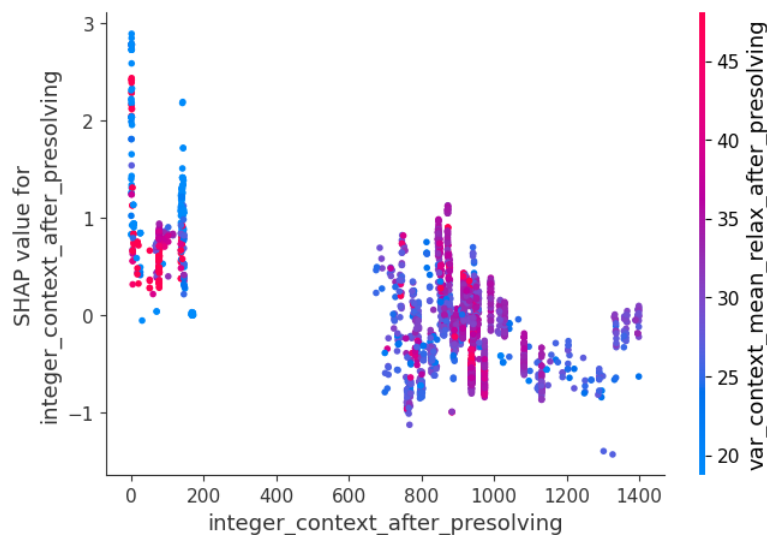


Рис. 4. График частичной зависимости признака `integer_context_after_presolving` от признака `var_context_mean_relax_after_presolving`

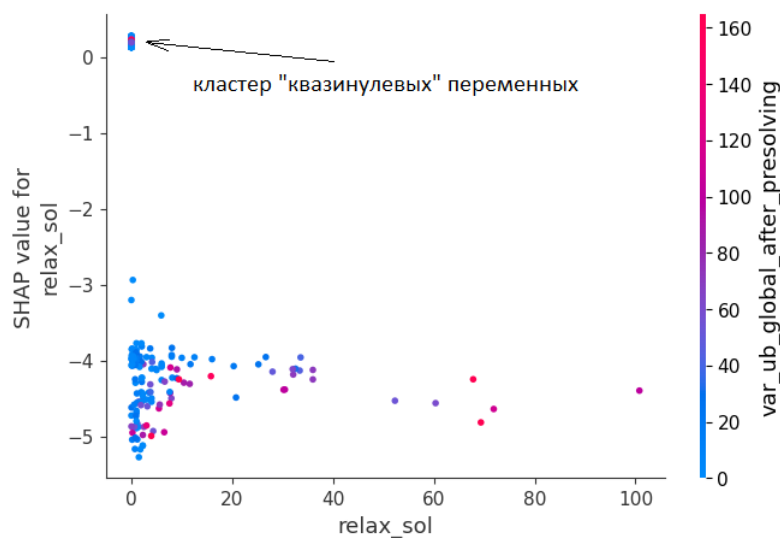


Рис. 5. График частичной зависимости признака `relax_sol` от признака `var_ub_global_after_presolving`

### 9.3. Прием подавления подгруппы первичных эвристик низкой эффективности

В некоторых случаях отдельные первичные эвристики могут оказаться не способными справиться со своей задачей, не оказывая никакого влияния на процедуру поиска решения, и все же потреблять предоставленные ресурсы.

Такие эвристики – условимся их называть первичными эвристиками низкой эффективности (ПЭНЭ) – можно выявить путем анализа статистической сводки `stat`-файла в разделе `Primal Heuristics`

Фрагмент файла статистической сводки `337_bin_default.stat`

...						
Primal Heuristics	:	ExecTime	SetupTime	Calls	Found	Best
LP solutions	:	0.00	-	-	0	0
relax solutions	:	0.00	-	-	0	0
pseudo solutions	:	0.00	-	-	0	0
...						

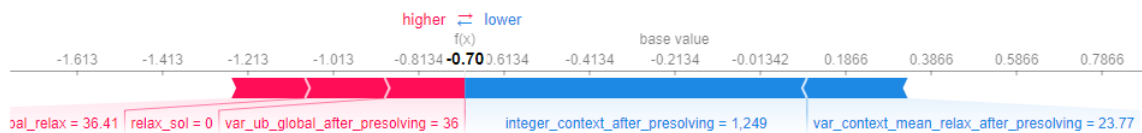


Рис. 6. Локальная интерпретация для первого экземпляра валидационного поднабора

conflictdiving	:	0.00	0.00	0	0	0	
crossover	:	0.00	0.00	0	0	0	
dins	:	0.00	0.00	0	0	0	
distributiondivin	:	0.00	0.00	0	0	0	
dualval	:	0.00	0.00	0	0	0	
farkasdiving	:	2032.89	0.00	1	0	0	# <- NB
feaspump	:	882.12	0.00	1	0	0	# <- NB
fixandinfer	:	0.00	0.00	0	0	0	
...							
intdiving	:	0.00	0.00	0	0	0	
intshifting	:	52.99	0.00	1	1	1	
...							

В данном случае ПЭНЭ являются `farkasdiving` и `feaspump`. Чтобы подавить эти эвристики при следующем запуске `SCIP`, достаточно включить следующие строки в конфигурационный файл `scip.set`<sup>5</sup>

`scip.set`

```
...
heuristics/farkasdiving/freq = -1
heuristics/feaspump/freq = -1
...
```

Доступ к статистической сводке можно получить либо в сессии `SCIP`, либо через одну из оберток над решателем (например, с помощью `PySCIPOpt`)

Фрагмент сессии `scip`. Получение статистической сводки

```
...
SCIP> read file.lp
SCIP> opt
SCIP> display stat
```

Получение статистической сводки через обертку `PySCIPOpt`

```
import pyscipopt

model = pyscipopt.Model()
model.readProblem("...")
model.readParams("...")
model.optimize()

model.printStatistics()
```

<sup>5</sup>При запуске интерактивной сессии через утилиту командной строки `scip`, решатель ищет этот файл в текущей директории и, если находит, автоматически вычитывает. При работе через `PySCIPOpt` требуется явно передавать путь до файла методу модели `readParams()`

## 9.4. Прием подбора порога бинаризации для бинарных переменных в релаксированном решении

Условимся *фиксацией* называть стратегию инициализации подгруппы переменных  $x_k$  (вещественных, бинарных или целочисленных), значения которых задаются на основе каких-либо эвристических соображений, например, касающихся специальных свойств матрицы ограничений, и способных в результате привести к такой постановке задачи, которую, используя механизмы первичных эвристик, сепараторов, пропагаторов и пр. можно развить в *допустимое целочисленное решение*.

Базовая идея построения *фиксации на бинарных переменных* заключается в том, чтобы значения бинарных переменных в релаксированном решении<sup>6</sup>  $\{r x_k^{(b)}\}_{k=1,\dots}$  интерпретировать как *степень уверенности* решателя в том, что рассматриваемую бинарную переменную можно выставить в единицу.

Если значение  $k$ -ой бинарной переменной  $r x_k^{(b)}$  превосходит некоторый *порог*  $\theta$ , то переменная выставляется в единицу, в противном случае – в ноль. Порог подбирается итерационно, начиная с некоторого нижнего значения  $\theta_l$  (по умолчанию  $\theta_l = 0$ ), увеличивая текущее значение порога на величину шага  $\Delta\theta$  и заканчивая верхним значением порога  $\theta_u$  (по умолчанию  $\theta_u = 1$ ).

Для практических целей достаточно остановиться на наименьшем значении порога  $\theta$ , который отвечает такой фиксации, которую решатель SCIP не отклоняет как неспособную привести к допустимому целочисленному решению.

Фрагмент лога решателя SCIP для случая фиксации, которую невозможно развить в допустимое целочисленное решение

```
...
SCIP Status      : problem is solved [infeasible]
Solving Time (sec) : 3.00
Solving Nodes    : 0
Primal Bound     : +1.0000000000000000e+20 (0 solutions)
Dual Bound      : +1.0000000000000000e+20
Gap              : 0.00 %
original problem has 740251 variables (2666 bin, 147789 int, 0 impl, 589796 cont) and 545350 constraints
...
```

После того как порог  $\theta$  подобран, бинарные переменные разбиваются на две подгруппы: подгруппу бинарных переменных, выставленных в ноль  $\{x_k^{(b_0)}\}$ , и подгруппу бинарных переменных, выставленных в единицу  $\{x_k^{(b_1)}\}$ . Долю бинарных переменных, выставленных в ноль обозначим через  $\delta_{b_0}$ , долю бинарных переменных, выставленных в единицу – через  $\delta_{b_1}$ , а целевую функцию, найденную при заданных долях – через  $f_\theta(\delta_{b_0}, \delta_{b_1})$ .

В результате получаем исследовательский инструмент, который дает возможность управлять решением через подбор долей  $\delta_{b_0}$  и  $\delta_{b_1}$  при найденном пороге  $\theta$ . Часто оказывается эффективным прием управления решением через подбор доли нулевых бинарных переменных  $\delta_{b_0}$ .

Целевая функция, вычисленная при единичной доле нулевых бинарных переменных  $f_\theta(\delta_{b_0} = 1)$ , как правило, значительно уступает целевой функции релаксированного решения  $f_r$ . Но тем не менее это решение может быть улучшено, сокращением доли  $\delta_{b_0}$  (см. рис. 7 и рис. 8).

<sup>6</sup>Верхний левый индекс « $r$ » указывает на релаксированное значение, а верхний правый « $(b)$ » – на то, что речь идет о бинарной переменной

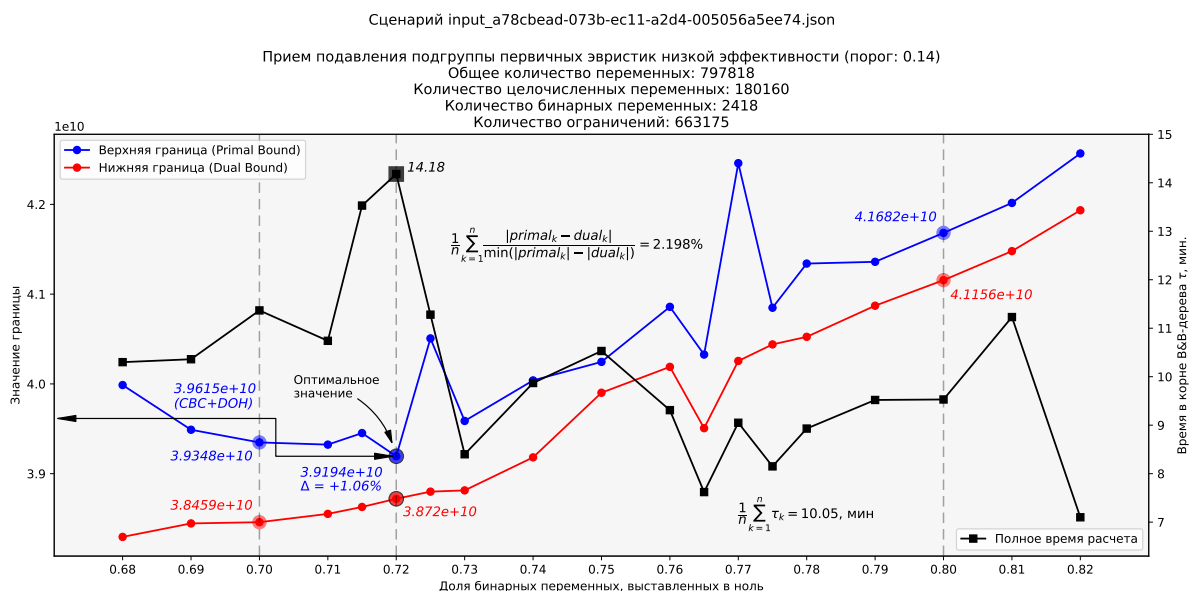


Рис. 7. Зависимость верхней границы решения от доли бинарных переменных, выставленных в ноль. Сценарий a78cbead

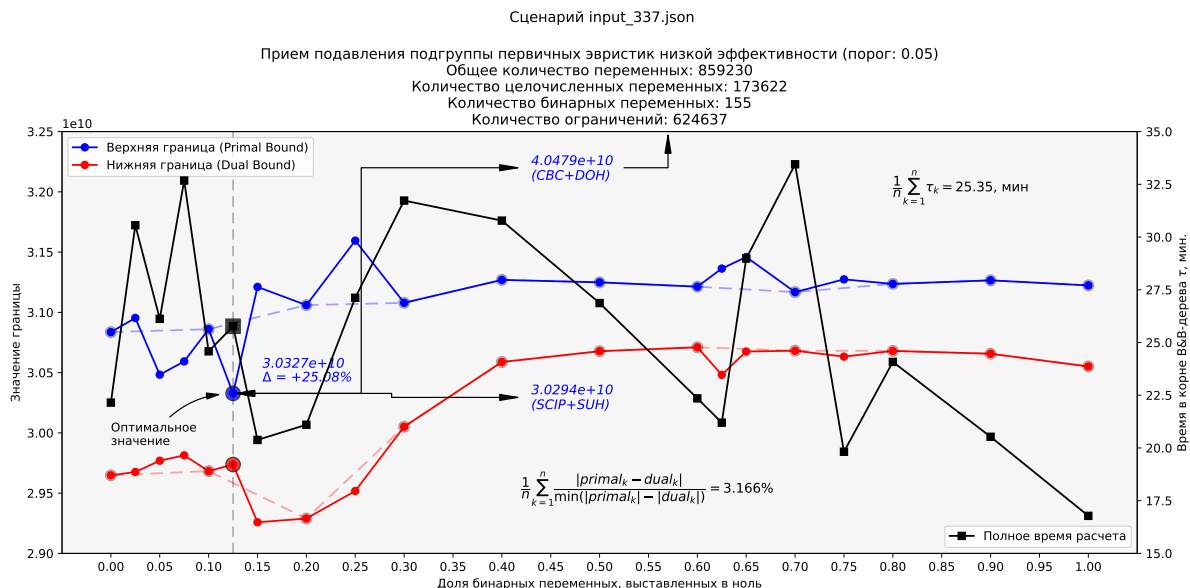


Рис. 8. Зависимость верхней границы решения от доли бинарных переменных, выставленных в ноль. Сценарий 337

Как видно из графиков, на кривой изменения верхней границы решения существует точка с наименьшим значением целевой функции  $f_{\theta}(\delta_{b_0})$  допустимого целочисленного решения. Эта точка и будет «оптимальной» для рассматриваемого сценария.

## 10. Методы машинного обучения в задачах комбинаторной оптимизации

### 10.1. Постановка задачи

Цель: Разработать процедуру построения частично-заданного решения на фиксациях для сценариев с матрицей ограничений произвольной структуры.

Вход: произвольная матрица ограничений<sup>7</sup>.

Выход: набор бинарных и целочисленных переменных, фиксация которых в ноль с высокой вероятностью приведет к допустимому целочисленному решению.

База: частично-заданное решение, построенное на фиксациях нулевых бинарных и целочисленных переменных в релаксированном решении.

## 10.2. Концепт матрицы признакового описания бинарных и целочисленных переменных

В качестве признаков бинарно-целочисленных переменных предлагается использовать:

1. *важный признак* Значение переменной  $x_i$  в «усредненном» релаксированном решении<sup>8</sup>,
2. Модифицированную Z-оценку на «усредненном» релаксированном решении,
3. *бесполезный признак* Дробную часть значения переменной  $x_i$  в «усредненном» релаксированном решении,
4. *важный признак* Пороги бинаризации на «усредненном» релаксированном решении (каждый порог это отдельный признак),
5. *важный признак* Число ограничений  $n_i$ , в которые входит рассматриваемая переменная  $x_i$ ,
6. *важный признак* Число положительных  $n_i^+$  и отрицательных  $n_i^-$  коэффициентов в ограничениях, ассоциированных с рассматриваемой переменной  $x_i$ ,
7. Булев маркер удаления переменной  $x_i$  после шага снижения размерности задачи,
8. *важный признак* Коэффициент  $c_i$  при переменной  $x_i$  в целевой функции  $\mathbf{c}^T \mathbf{x}$ ,
9. *бесполезный признак* Вероятность<sup>9</sup> того, что  $i$ -ая бинарная или целочисленная переменная  $x_i$  будет выставлена в 1 (индекс « $-i$ » означает без учета  $i$ -ой переменной)

$$\mathbf{P}(x_i = 1) = \sigma\left(\frac{1}{t} (\mathbf{c}^T \mathbf{x})_{-i}\right),$$

где  $\sigma$  – логистический сигмоид,  $t$  – «температура» (чем выше температура, тем случайнее выход),  $\mathbf{c}$  – вектор коэффициентов целевой функции,  $\mathbf{x}$  – вектор значений переменных в релаксированном решении.

10. Важность  $x_i$  переменной с точки зрения пресолверов.

## 10.3. Стратегии решения задачи

### 10.3.1. Стратегия №1. Обнаружение аномалий

Задачу построения частично-заданного решения на фиксациях предлагается свести к задаче обнаружения аномалий в данных. Бинарные и целочисленные переменные, которые как ожидается примут *нулевые значения* в допустимом целочисленном решении будем считать «штатным» режимом, а бинарные и целочисленные переменные, которые как ожидается примут *ненулевые значения* в допустимом целочисленном решении – аномалиями. Такие «аномальные» экземпляры остаются без рекомендуемого значения для фиксации, а оставшиеся нулевые «штатные»

<sup>7</sup>Предполагается, что матрица ограничений имеет низкую меру обусловленности

<sup>8</sup>Задача линейного программирования в релаксированной постановке решается с использованием различных методов (двойственный симплекс-метод, метод внутренней точки и т.д.), а затем полученные решения усредняются

<sup>9</sup>Идея построения признака основана на способе вычисления вероятности единичного выхода нейрона в машинах Больцмана [2, стр. 653]



бинарные и целочисленные переменные фиксируются в ноль и на этом процедура построения частично-заданного решения считается завершенной.

Для повышения надежности прогноза предлагается использовать ансамбль детекторов аномалий. Решение о фиксации бинарной или целочисленной переменной в ноль принимается на основании большинства голосов ансамбля детекторов.

Набор данных представляет собой неупорядоченную коллекцию матриц признакового описания, ассоциированных с соответствующими *lp/mps*-файлами математической постановки задачи (условимся называть их *сценариями*).

Ансамбль детекторов аномалий обучается по роторной схеме:

- На  $i$ -ой итерации все *матрицы признакового описания* (всего в наборе  $S$  матриц/сценариев) кроме  $i$ -ой матрицы используются для обучения детекторов, а на  $i$ -ой матрице признакового описания строится прогноз аномальных экземпляров, которые помечаются как «-1». В результате получается коллекция бинарных и целочисленных переменных, помеченных либо как «0», либо как «-1». Построенное решение сравнивается с допустимым целочисленным решением с помощью различных метрик качества (параметрическое гармоническое среднее, капша Коэна, коэффициент корреляции Метьюса и т.д.). Вычисленные для  $i$ -ой матрицы метрики качества и построенное частично-заданное решение на фиксациях сохраняются в директории результатов,
- Затем описанный шаг повторяется для оставшихся матриц признакового описания объекта. По окончании процедуры для каждого сценария:
  - будут вычислены метрики качества,
  - будет построено частично-заданное решение на фиксациях,

Полученные частично-заданные решения на фиксациях подаются на вход решателю SCIP. Если SCIP удалось найти решение, обозначаемое как  $s_{ML}$ , то оно сравнивается с решением  $s_{FZB}$ , полученным с помощью метаконфигурации FZBIVSUNPB (см. подраздел 11.2), по времени работы и по значению верхней границы решения.

---

#### Замечание

Как правило, в задачах обнаружения аномалий не выполняют подбор гиперпараметров детектора, но в данном случае кажется полезным изучить поведение детектора хотя бы в зависимости от параметра контаминации. Дело в том, что на практике эффективность детектора может существенно изменяться в зависимости от значений управляющих параметров

---

На всех сценариях группы ИКП (см. раздел 11) обнаруживается серьезный дисбаланс экземпляров положительного («аномалии», ненулевые значения переменных) и отрицательного («штатные» экземпляры, нулевые значения переменных) классов. Ожидается, что эффективность модели машинного обучения главным образом будет зависеть от способности модели выявлять аномальные экземпляры.

Действительно, *ошибка первого рода* (ложное срабатывание, т.е. когда отрицательный «штатный» экземпляр принимается за «аномальный» положительный) приводит к тому, что нулевая переменная *не будет* зафиксирована в ноль в частично-заданном решении, что с высокой вероятностью снизит производительность решателя SCIP.

Тогда как *ошибка второго рода* (пропуск объекта, т.е. когда «аномальный» положительный экземпляр принимается за «штатный» отрицательный) приводит к тому, что ненулевая переменная в частично-заданном решении будет зафиксирована в ноль. Это сделает частично-заданное решение не способным развиваться в допустимое целочисленное, что значительно хуже.

Таким образом, кажется разумным сосредоточить усилия на том, чтобы минимизировать ошибку второго рода, и в результате свести к минимуму число пропусков аномалий.

Проще всего оценить качество модели с учетом большего влияния ошибок второго рода с помощью  $F_\beta$ -меры при значениях параметра  $\beta > 1$

$$F_\beta = (1 + \beta^2) \frac{\text{precision} \cdot \text{recall}}{\beta^2 \text{precision} + \text{recall}},$$

где precision – точность, recall – полнота.

---

#### Замечание

Провести анализ приема подбора порога бинаризации. И проработать схему подбора гиперпараметров детекторов

---

**Анализ производительности методов обнаружения аномалий** Рекомендуемые значения некоторых гиперпараметров для детекторов некоторых семейств звучат следующим образом [3]:

- для KNN (k Nearest Neighbors<sup>10</sup>) и LOF (Local Outlier Factor):  $k = \max(10; 0.03 |D|)$ , где  $|D|$  – число экземпляров в наборе данных,
- для HBOS (Histogram-based Outlier Score):  $n\_bins = \sqrt{|D|}$ ,
- для IForest (Isolation Forest): число деревьев  $n\_estimators=100$  и число экземпляров на дерево  $max\_samples=256$ ,
- для CBLOF (Clustering-Based Local Outlier Factor):  $\alpha = 0.90$ ,  $\beta = 5$  и  $k = 10$ ,
- для OCSVM (One-Class Support Vector Machines): ядро RFB( $\nu = 0.5$ ,  $\gamma = 1/m$ ), где  $m$  – число признаков в наборе данных  $D$ .

Перечисленные ниже детекторы показали крайне низкую производительность на сценариях группы ИКП:

- KNN,
- Feature Bagging,
- ABOD (Angle-Based Outlier Detection using approximation)/FastABOD,
- LOCI (Fast outlier detection using the local correlation integral),
- CBLOF (Clustering-Based Local Outlier Factor): достаточно быстрый, но результаты отвратительные (очень низкие значения ключевых метрик качества),
- XGBOOST<sup>11</sup> (Extreme Boosting Based Outlier Detection): безумно медленный<sup>12</sup>,
- R-Graph (Outlier detection by R-graph).

Главный детектор аномалий предлагается строить с помощью агрегатора SUOD<sup>13</sup> (Accelerating Large-scale Unsupervised Heterogeneous Outlier Detection) на следующих базовых детекторах:

- ECOD (Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions),
- COPOD (Copula-Based Outlier Detection),
- IForest (Isolation Forest),
- HBOS (Histogram-based Outlier Score).

---

<sup>10</sup>Расстояние от  $k$ -ого ближайшего соседа рассматривается как мера аномальности экземпляра

<sup>11</sup>Требует разметки

<sup>12</sup>В <https://github.com/yzhao062/pyod/issues/152> рекомендуется использовать SUOD

<sup>13</sup><https://www.andrew.cmu.edu/user/yuezhao2/papers/21-mlsys-suod.pdf>

### 10.3.2. Стратегия №2. Бинарная классификация со слабо выраженным миноритарным классом

Задачу построения частично-заданного решения на фиксациях предлагается свести к задаче бинарной классификации со слабо выраженным миноритарным классом (данные с сильным дисбалансом).

*Раздел в разработке ...*

## 10.4. Трансфер выявленного паттерна. Сценарии группы СОП

Условимся *трансфером выявленного паттерна* (или просто *трансфером паттерна*) называть явление, состоящее в том, что модель, обученная на сценариях одной группы (сценарии обучающего поднабора), оказывается способной строить корректные прогнозы на сценариях другой группы (сценарии тестового поднабора), обладающих четкими дискриминирующими атрибутами (структурные особенности матрицы ограничений и пр.), которые позволяют с высокой степенью уверенности отделять сценарии обучающего поднабора от сценариев тестового поднабора.

Другими словами, в отличие от классической постановки машинного обучения – в которой экземпляры обучающего и тестового поднаборов данных должны быть похожи друг на друга – в данном случае модель машинного обучения предлагается обучать и тестировать на сценариях, которые значительно отличаются друг от друга по каким-то ключевым атрибутам.

### 10.4.1. Сценарий tmpfvprqodxw.lp без бинарных переменных

Исследование вопроса о трансфере паттерна начнем с рассмотрения простого сценария группы СОП tmpfvprqodxw.lp <https://disk.yandex.ru/d/K7bvClpltotqlg>, а обучать модель машинного обучения будем в соответствии со стратегией №1 (стр. 24).

В случае сценария tmpfvprqodxw.lp для простоты можно ограничиться рассмотрением только детектора HBOS (без агрегации прогнозов других детекторов с помощью обертки SUOD) и обучать его на сценарии группы ИКП f398266b\_bin.lp (см. раздел 11.2.4).

Для того чтобы использовать не ансамбль детекторов аномалий, а лишь какой-то конкретный детектор, достаточно в конфигурационном файле main\_config.yaml передать полю use детектора значение False

main\_config.yaml. Использовать только детектор HBOS

```
...
detector_config:
  # Строит ансамбль детекторов аномалий
  SUOD: # Scalable Unsupervised Outlier Detection https://www.andrew.cmu.edu/user/yuezhao2/
        papers/21-mlsys-suod.pdf
        use: !!bool False # <--- NB
        # Допустимые значения 'combination': average, maximization
        combination: !!str average # стратегия агрегации прогнозов ансамбля детекторов
        contamination: !!float 0.10 # доля выбросов в наборе данных; принимает значения из диапазон
        а (0.0; 0.5)
        n_jobs: -1 # число параллельно выполняемых задач
        verbose: !!bool True # флаг подробного вывода информации о построении модели
  # Перечень детекторов для SUOD-ансамбля. Если SUOD.use=True, то перечисленные ниже детекторы,
  # у которых
  # атрибут DETECTOR.use=True, будут добавлены в список SUOD().base_estimators.
  # Если SUOD.use=False, то поиск аномалий будет выполняться с помощью одного из приведенных ниж
  # е детекторов,
```

```

# у которого атрибут DETECTOR.use=True
COPOD: # Copula Based Outlier Detector
  use: !!bool False # <--- NB
  contamination: !!float 0.10 # доля выбросов в наборе данных; принимает значения из диапазон
  а (0.0; 0.5)
  n_jobs: -1 # число параллельно выполняемых задач
ECOD: # Unsupervised Outlier Detection Using Empirical Cumulative Distribution Functions
  use: !!bool False # <--- NB
  contamination: !!float 0.10 # доля выбросов в наборе данных; принимает значения из диапазон
  а (0.0; 0.5)
  n_jobs: -1 # число параллельно выполняемых задач
IForest: # Wrapper of scikit-learn Isolation Forest with more functionalities
  use: !!bool False # <--- NB
  n_estimators: !!int 250 # число деревьев принятия решений в лесе
  contamination: !!float 0.10 # доля выбросов в наборе данных; принимает значения из диапазон
  а (0.0; 0.5)
  n_jobs: -1 # число параллельно выполняемых задач
HBOS: # Histogram-based outlier detection
  use: !!bool True # <--- NB
  n_bins: !!int 10 # число бинов для построения гистограммы
  alpha: !!float 0.05 # параметр регуляризации
  contamination: !!float 0.10 # доля выбросов в наборе данных; принимает значения из диапазон
  а (0.0; 0.5)

```

Приведенный на рис. 9 график показывает, что

- о настройки решателя SCIP, ответственные за выбор переменных при ветвлении<sup>14</sup> и разрешении конфликтов<sup>15</sup>, а также прием подавления подгруппы первичных эвристик низкой эффективности помогают снизить временные издержки при незначительном ухудшении целевой функции (зеленая кривая) относительно решения, полученного с помощью решателя SCIP с настройками по умолчанию (красная кривая),
- о дополнительное снижение временных затрат можно получить подбором гиперпараметров детектора<sup>16</sup> (синяя кривая).

Детектору аномалий HBOS с подбором параметра контаминации (`contamination=0.04`)<sup>17</sup> удалось снизить количество бинарных переменных – на 98, ограничений – на 177, а временные издержки снизились в 2.38 раза.

#### 10.4.2. Синтетический сценарий 1664182546\_82382.lp с бинарными переменными

##### Статистика<sup>18</sup>

Общее количество переменных: 5100 (4123)

Количество целочисленных переменных: 0 (0)

Количество бинарных переменных: 1768 (1132)

Количество ограничений: 11193 (10461)

lp-файл: <https://disk.yandex.ru/d/FuEBWt4zvFIseA>

Блок подавления подгруппы первичных эвристик низкой эффективности конфигурационного файла SCIP

<sup>14</sup>Параметр `branching/preferbinary`

<sup>15</sup>Параметр `conflict/preferbinar`

<sup>16</sup>В данном случае подбирался только гиперпараметр контаминации

<sup>17</sup>В библиотеке PyOD все детекторы аномалий имеют контаминацию уровня 0.10

<sup>18</sup>В скобках указана размерность задачи после шага пресолвинга с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов

Сценарий tmpfvpqodxw.lp (обучение на f398266b\_bin.lp)

Общее количество переменных: 5308  
Количество целочисленных переменных: 1197  
Количество бинарных переменных: 0  
Количество ограничений: 13065

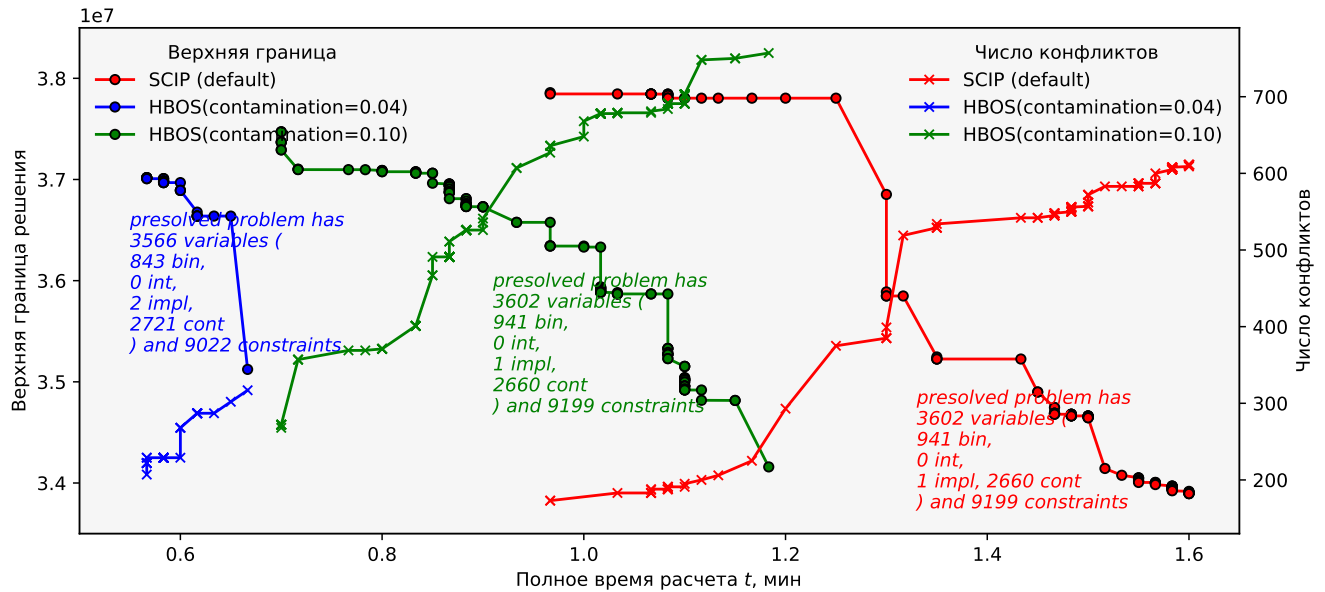


Рис. 9. Сводка результатов вычислительных экспериментов на сценарии группы СОП tmpfvpqodxw.lp

Фрагмент файла scip.set. Сценарий 1664182546\_82382.lp

```
...
heuristics/adaptivediving/freq = -1
heuristics/fracdiving/freq = -1
heuristics/linesearchdiving/freq = -1
heuristics/objpscostdiving/freq = -1
heuristics/pscostdiving/freq = -1
heuristics/rootsoldiving/freq = -1
heuristics/veclendiving/freq = -1
```

На рис. 10 приведены результаты сравнительного анализа запусков i) решателя SCIP с настройками по умолчанию, ii) решателя SCIP на частично-заданном решении, полученном с помощью ансамбля детекторов аномалий, и iii) решателя SCIP на фиксации, подготовленной с помощью изолированного детектора HBOS.

Как видно из рисунка, решатель SCIP с настройками по умолчанию (синяя кривая) первое допустимое целочисленное решение с адекватным зазором находит гораздо позже схемы на частично-заданном решении, полученном с помощью ансамбля детекторов (красная кривая). Однако, спустя некоторое время схема с настройками по умолчанию быстрее выходит на конкурентное значение целевой функции (41389.75 против 41557.30).

Схема с подбором гиперпараметра контаминации изолированного детектора HBOS, несмотря на то, что размерность задачи снижается, приводит к очень слабому решению.

**Вывод по сценарию:** принимая во внимание, что ансамбль детекторов аномалий обучался лишь на одном сценарии группы ИКП, который существенно и с точки зрения размерности, и с точки зрения структуры матрицы ограничений отличается от сценария, на котором строился

прогноз модели, допустимо говорить о трансфере/переносе шаблона, выявленного на сценарии f398266b\_bin.lp группы ИКП.

Сценарий 1664182463\_82382.lp (обучение на f398266b\_bin.lp)

Общее количество переменных: 5100  
Количество целочисленных переменных: 0  
Количество бинарных переменных: 1768  
Количество ограничений: 11193

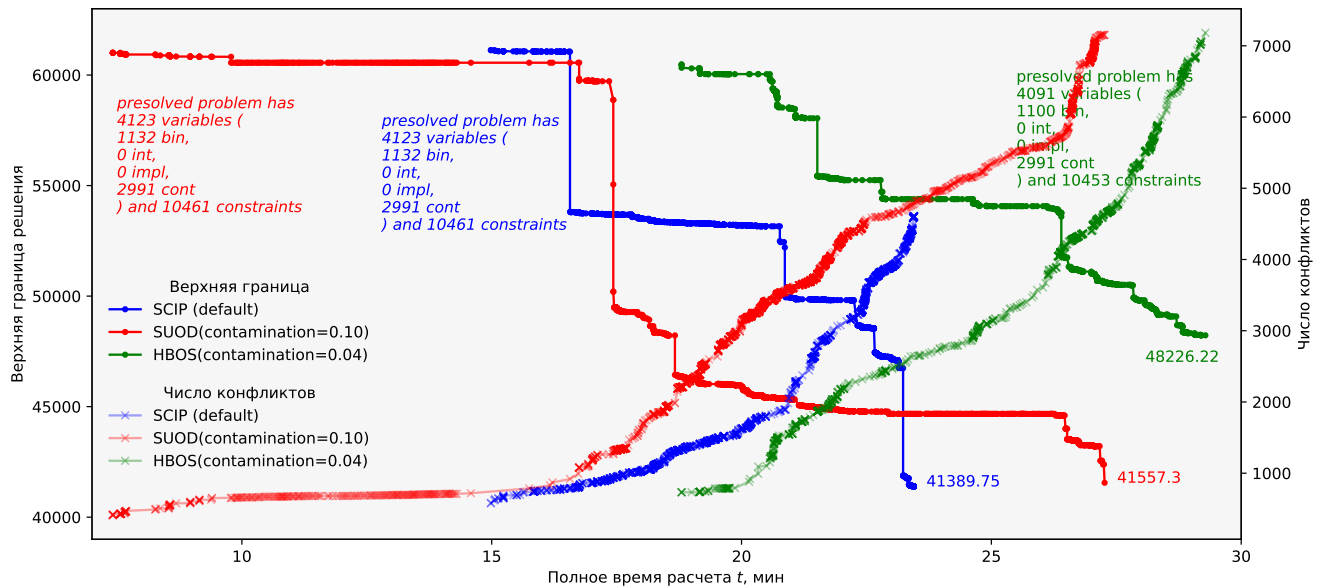


Рис. 10. Сводка результатов вычислительных экспериментов на сценарии группы СОП 1664182546\_82382.lp

#### 10.4.3. Синтетический сценарий 1664182533\_1587787.lp с бинарными переменными

##### Статистика<sup>19</sup>

Общее количество переменных: 4759 (3780)

Количество целочисленных переменных: 0 (0)

Количество бинарных переменных: 1701 (1063)

Количество ограничений: 10307 (9581)

lp-файл: <https://disk.yandex.ru/d/n0Dqn6pr6GK9mg>

Блок подавления подгруппы первичных эвристик низкой эффективности конфигурационного файла SCIP

Фрагмент файла scip.set. Сценарий 1664182533\_1587787.lp

```
...
heuristics/adaptivediving/freq = -1
heuristics/fracdiving/freq = -1
heuristics/linesearchdiving/freq = -1
heuristics/objpscostdiving/freq = -1
heuristics/pscostdiving/freq = -1
heuristics/rootsoldiving/freq = -1
heuristics/veclendiving/freq = -1
```

<sup>19</sup>В скобках указана размерность задачи после шага пресолвинга с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов

На рис. 11 приведены результаты сравнительного анализа запусков i) решателя SCIP с настройками по умолчанию, ii) решателя SCIP на частично-заданном решении, полученном с помощью ансамбля детекторов аномалий, и iii) решателя SCIP на фиксации, подготовленной с помощью изолированного детектора HBOS.

Здесь схема с настройками по умолчанию проигрывает схеме на частично-заданном решении, построенном с помощью ансамбля детекторов аномалий, и по времени расчета, и по значению целевой функции. Подбор параметра контаминации детектора HBOS как и в предыдущем случае не позволяет улучшить решение – кривая «замирает» на асимптоте 52070.46.

Таким образом, в данном случае ансамбль детекторов аномалий с обреткой SUOD снижает временные издержки на получение решения и одновременно улучшает целевую функцию.

**Вывод по сценарию:** принимая во внимание, что ансамбль детекторов аномалий обучался лишь на одном сценарии группы ИКП, который существенно и с точки зрения размерности, и с точки зрения структуры матрицы ограничений отличается от сценария, на котором строился прогноз модели, допустимо говорить о трансфере/переносе шаблона, выявленного на сценарии f398266b\_bin.lp группы ИКП.

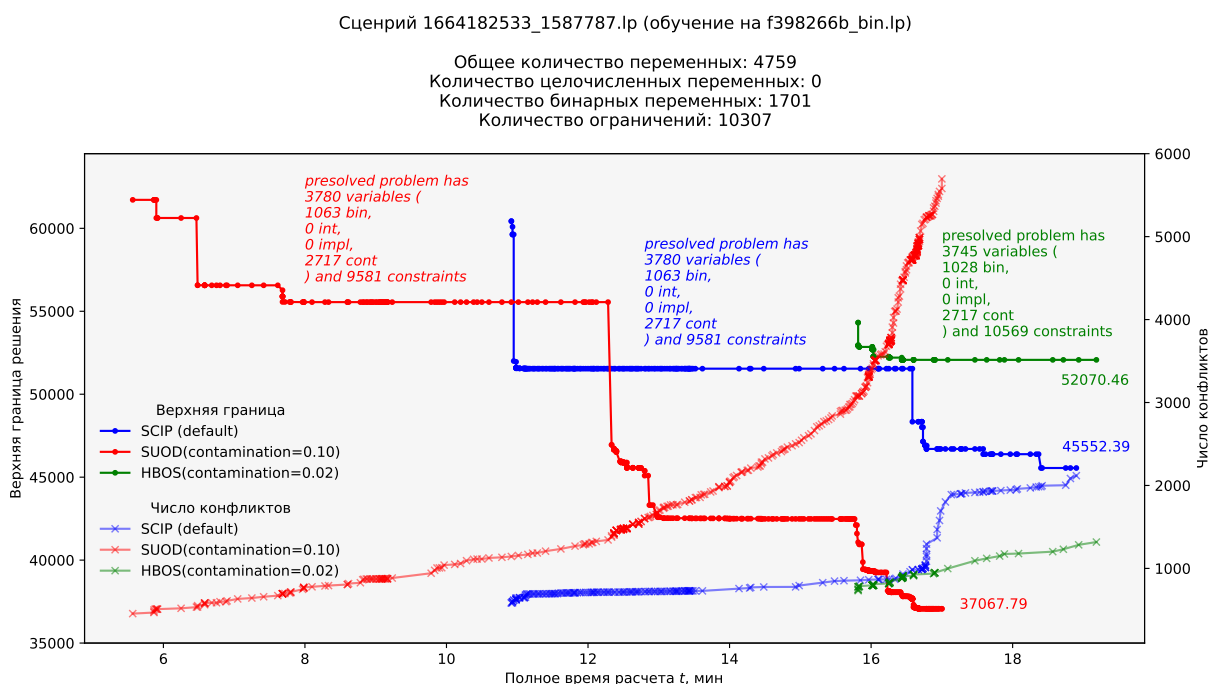


Рис. 11. Сводка результатов вычислительных экспериментов на сценарии группы СОП 1664182533\_1587787.lp

#### 10.4.4. Синтетический сценарий 1664182480\_4326847.lp с бинарными переменными

##### Статистика<sup>20</sup>

Общее количество переменных: 7123 (6445)

Количество целочисленных переменных: 0 (0)

Количество бинарных переменных: 1548 (1324)

Количество ограничений: 17696 (16805)

<sup>20</sup>В скобках указана размерность задачи после шага пресолвинга с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов



lp-файл: [https://disk.yandex.ru/d/f\\_6GH9mzzxAGQg](https://disk.yandex.ru/d/f_6GH9mzzxAGQg)

Блок подавления подгруппы первичных эвристик низкой эффективности конфигурационного файла SCIP

Фрагмент файла scip.set. Сценарий 1664182480\_4326847.lp

```
...
heuristics/adaptivediving/freq = -1
heuristics/fracdiving/freq = -1
heuristics/linesearchdiving/freq = -1
heuristics/objpscostdiving/freq = -1
heuristics/pscostdiving/freq = -1
heuristics/rootsoldiving/freq = -1
heuristics/veclendiving/freq = -1
```

На рис. 12 приведены результаты сравнительного анализа запусков i) решателя SCIP с настройками по умолчанию и ii) решателя SCIP на фиксации, подготовленной с помощью изолированного детектора HBOS.

На рассматриваемом сценарии получить решение с помощью ансамбля детекторов аномалий за отведенное для поиска время не удалось, однако, изолированный детектор HBOS с подобранным параметром контаминации смог выйти на значение целевой функции 53682.08. Это решение проигрывает решению, полученному с помощью SCIP базовой конфигурации (47245.97), но тем не менее указывает жизнеспособность концепции использования стратегии обнаружения аномалий для построения частично-заданного решения на фиксациях с подбором параметра контаминации детекторов.

**Вывод по сценарию:** принимая во внимание, что ансамбль детекторов аномалий обучался лишь на одном сценарии группы ИКП, который существенно и с точки зрения размерности, и с точки зрения структуры матрицы ограничений отличается от сценария, на котором строился прогноз модели, допустимо говорить о трансфере/переносе шаблона, выявленного на сценарии f398266b\_bin.lp группы ИКП.

#### 10.4.5. Синтетический сценарий 1664182523\_380519.lp с бинарными переменными

##### Статистика<sup>21</sup>

Общее количество переменных: 4578

Количество целочисленных переменных: 0

Количество бинарных переменных: 1331

Количество ограничений: 10722

lp-файл: <https://disk.yandex.ru/d/i-FhZ9LD8ToeXg>

Блок подавления подгруппы первичных эвристик низкой эффективности конфигурационного файла SCIP

Фрагмент файла scip.set. Сценарий 1664182523\_380519.lp

```
...
heuristics/adaptivediving/freq = -1
heuristics/fracdiving/freq = -1
heuristics/linesearchdiving/freq = -1
heuristics/objpscostdiving/freq = -1
```

<sup>21</sup>В скобках указана размерность задачи после шага пресолвинга с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов



Сценарий 1664182480\_4326847.lp (обучение на f398266b\_bin.lp)

Общее количество переменных: 7123  
Количество целочисленных переменных: 0  
Количество бинарных переменных: 1548  
Количество ограничений: 17696

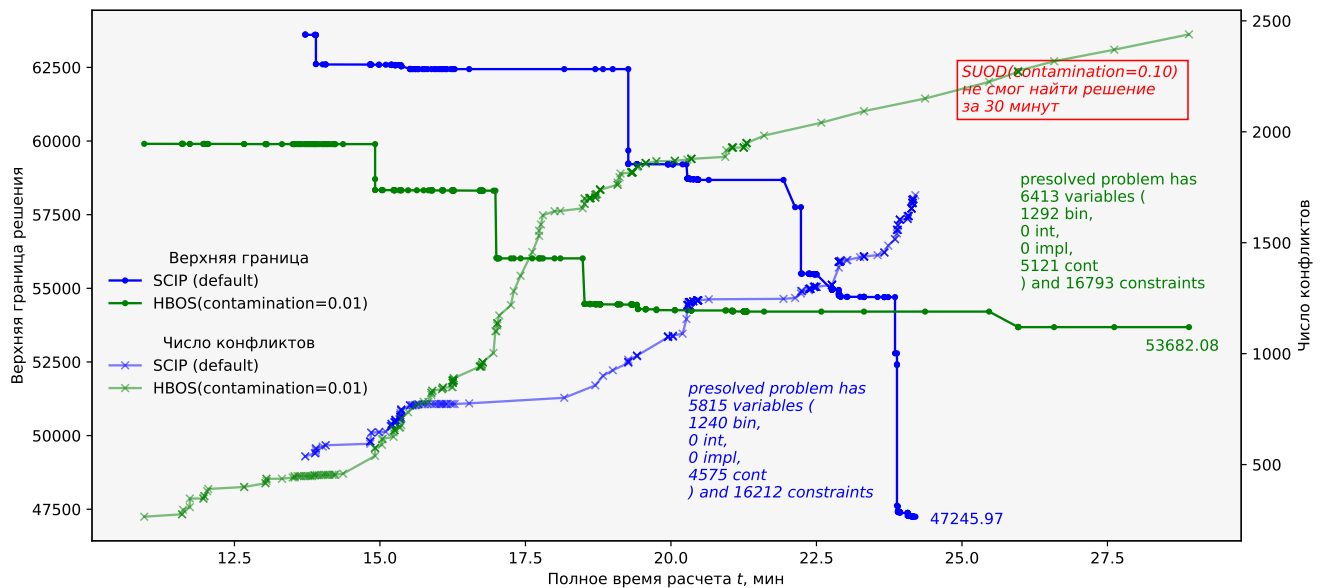


Рис. 12. Сводка результатов вычислительных экспериментов на сценарии группы СОП 1664182480\_4326847.lp

```
heuristics/pscostdiving/freq = -1
heuristics/rootsoldiving/freq = -1
heuristics/veclending/freq = -1
```

На рис. 13 приведены результаты сравнительного анализа запусков i) решателя SCIP с настройками по умолчанию, и ii) решателя SCIP на частично-заданном решении, полученном с помощью ансамбля детекторов аномалий.

Здесь ансамбль детекторов аномалий выигрывает 2.78 минуты при целевой функции, значение которой практически не отличается от значения целевой функции в решении, полученном с помощью решателя SCIP базовой конфигурации.

**Вывод по сценарию:** принимая во внимание, что ансамбль детекторов аномалий обучался лишь на одном сценарии группы ИКП, который существенно и с точки зрения размерности, и с точки зрения структуры матрицы ограничений отличается от сценария, на котором строился прогноз модели, допустимо говорить о трансфере/переносе шаблона, выявленного на сценарии f398266b\_bin.lp группы ИКП.

## 10.5. Концепт построения матрицы признакового описания объекта для задачи выбора стратегии поиска решения

Каждую проблему из набора решаем различными стратегиями (HiGHS с настройками по умолчанию, SCIP с настройками по умолчанию, ZuoOpt 1-ой конфигурации, ZuoOpt 2-ой конфигурации и т.д.). Затем выбираем наиболее эффективную стратегию для каждой проблемы. Это и будет разметка. Проблему отображаем на вектор-строку матрицы признакового описания объекта и в итоге получаем матрицу плана. Теперь требуется решить задачу мультиклассовой классификации. Классификатор будет возвращать вероятность принадлежности экземпляра классам.

Сценрий 1664182523\_380519.lp (обучение на f398266b\_bin.lp)

Общее количество переменных: 4578  
Количество целочисленных переменных: 0  
Количество бинарных переменных: 1331  
Количество ограничений: 10722

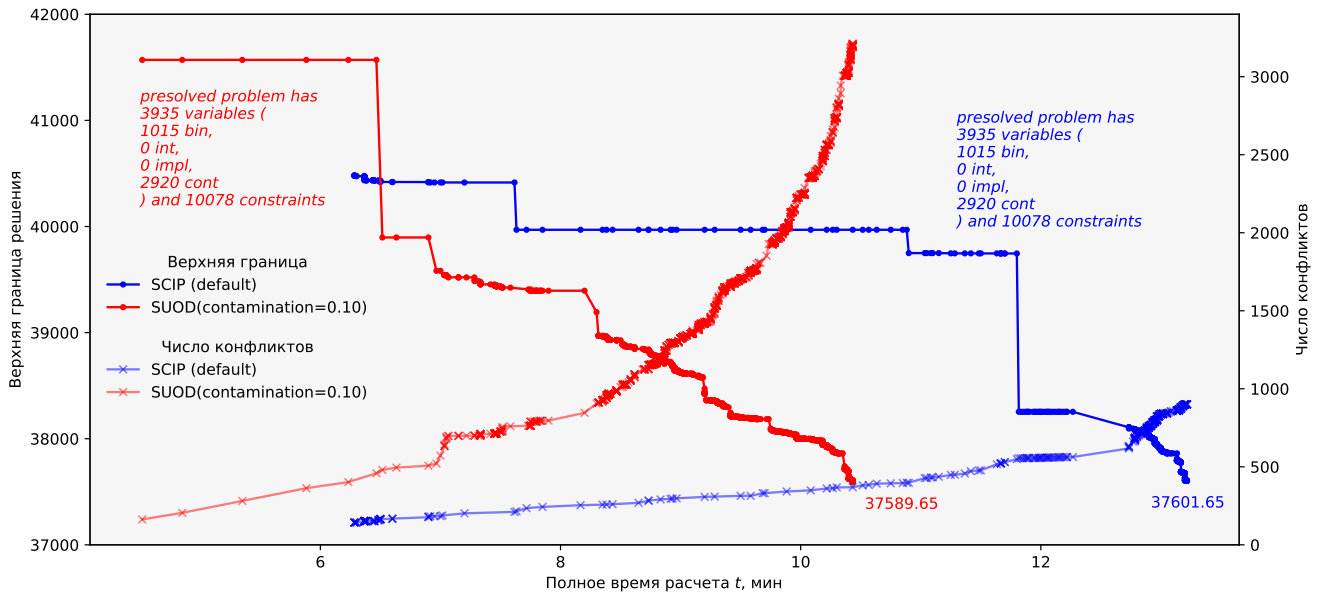


Рис. 13. Сводка результатов вычислительных экспериментов на сценарии группы СОП 1664182523\_380519.lp

Выбираем класс с наибольшей вероятностью. Таким образом можно связать стратегию поиска решения с представлением проблемы и проранжировать стратегии. Начинаем с самой эффективной, а потом по убывающей.

## 11. Описание вычислительных экспериментов на сценариях группы ИКП

На всех сценариях группы ИКП (как с бинарными переменными, так и без них) решения удавалось найти с помощью *метаконфигурации* (см. раздел 11.2), включающей прием подавления подгруппы первичных эвристик низкой эффективности и процедуру построения частично-заданного решения на фиксациях (для нулевых бинарных и целочисленных переменных).

### 11.1. Поиск решения на сценариях без бинарных переменных.

#### Метаконфигурации SUH, FZBIVSUHPB и ансамбль детекторов аномалий

Метаконфигурация<sup>22</sup> SUH (Suppress Useless Heuristics) процедуры поиска решения сводится к приему подавления подгруппы первичных эвристик низкой эффективности.

---

#### Замечание

Решение получено без доменно-ориентированных эвристик, «теплого» старта и подбора параметров решателя

---

<sup>22</sup>Под метаконфигурацией понимается совокупность конфигурации решателя и набора эвристических приемов

Конфигурация решателя SCIP для всех сценариев группы ИКП (без бинарных переменных) имеет вид

scip.set. Сценарии группы ИКП без бинарных переменных

```
# критерии останова и перезапуска
limits/time = 7200
limits/gap = 0.02 # решение останавливается при зазоре <= 2%

# подавление подгруппы первичных эвристик низкой эффективности
heuristics/farkasdiving/freq = -1
heuristics/feaspump/freq = -1
heuristics/randrounding/freq = -1
heuristics/shiftandpropagate/freq = -1
heuristics/shifting/freq = -1
```

Сводка результатов вычислительных экспериментов доступна по ссылке <https://docs.google.com/document/d/1V9fZLT9cXkbVQ5BvMCwzKrAiASZ2v4-01Z68jVBZUBU/edit?usp=sharing>.

#### 11.1.1. Сценарий F398266B без бинарных переменных

##### Статистика

Общее количество переменных: 774901

Количество целочисленных переменных: 172449

Количество бинарных переменных: 0

Количество ограничений: 650263

lp-файл: [https://disk.yandex.ru/d/o\\_eAb9475u5ueg](https://disk.yandex.ru/d/o_eAb9475u5ueg)

##### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- o INTSHIFING,
- o RENS.

Файл решения задачи (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/URRnZ8soTaJEgQ>

Файл статистической сводки (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/N2tfhj1N6Rczza>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/-y7p5FyJyYirkw>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/1JaMC9aFjubDbA>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 1.063% лучше в смысле целевой функции и на 10.20% – в смысле временных издержек (рис. 14).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 1.155% лучше в смысле целевой функции и на 65.27% – в смысле временных издержек (табл. 1).

Синим цветом обозначен выигрыш в процентах.

Таблица 1. Сводка результатов анализа эффективности метаконфигураций SUN и FZBIVSUNPB. Сценарий f398266b без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	21.38	5.905048
SCIP+SUN	19.27 +9.87%	5.842154 +1.065%
SCIP+FZB...	9.43 +55.89%	5.836815 +1.155%

Сценарий input\_f398266b-093b-ec11-a2d4-005056a5ee74.json

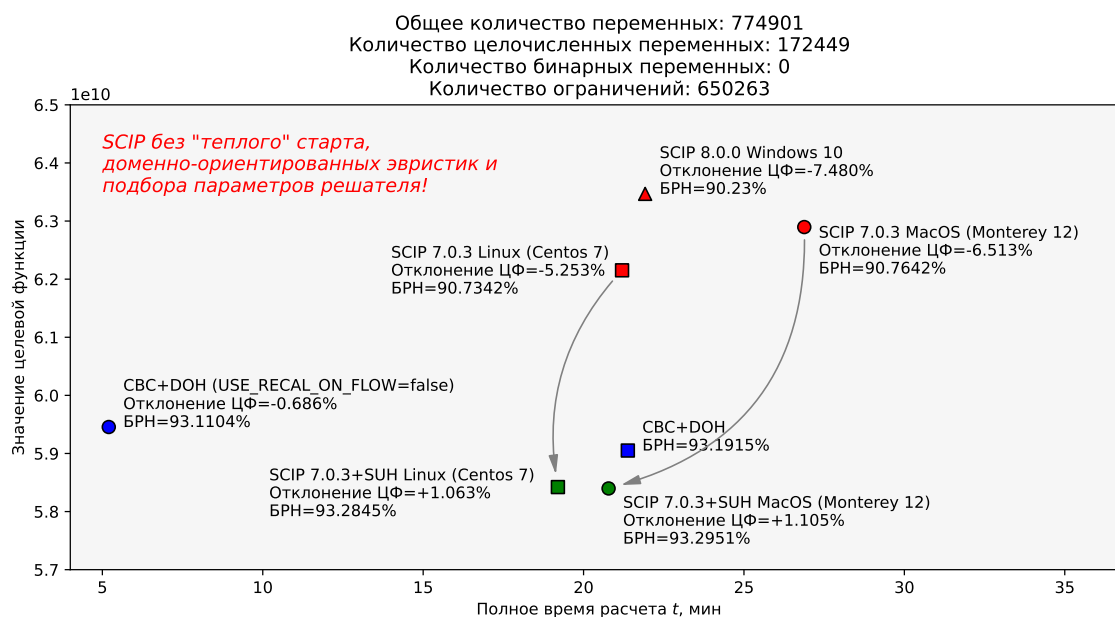


Рис. 14. Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий f398266b без бинарных переменных

### 11.1.2. Сценарий 50197DF7 без бинарных переменных

#### Статистика

Общее количество переменных: 718464

Количество целочисленных переменных: 159332

Количество бинарных переменных: 0

Количество ограничений: 595797

lp-файл: [https://disk.yandex.ru/d/K0\\_xj9dkgUdcog](https://disk.yandex.ru/d/K0_xj9dkgUdcog)

#### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

Файл решения задачи (метаконфигурация SUN) доступен по ссылке <https://disk.yandex.ru/d/R4B1fkTx-nE3tg>

Файл статистической сводки (метаконфигурация SUN) доступен по ссылке <https://disk.yandex.ru/d/BLvUmZ43vtMFKg>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/yMFLr-6mLfdPAw>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/XiRSvteL9xC4pg>

**Вывод по сценарию:** описанная выше метаконфигурация SUN приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 1.25% лучше в смысле целевой функции и на 46.43% – в смысле временных издержек (рис. 15).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 1.191% лучше в смысле целевой функции и на 82.13% – в смысле временных издержек (табл. 2).

Синим цветом обозначен выигрыш в процентах.

Таблица 2. Сводка результатов анализа эффективности метаконфигураций SUN и FZBIVSUHPB. Сценарий 50197df7 без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	18.35	3.585532
SCIP+SUN	9.83 +46.43%	3.540567 +1.252%
SCIP+FZB...	3.28 +82.13%	3.542843 +1.191%

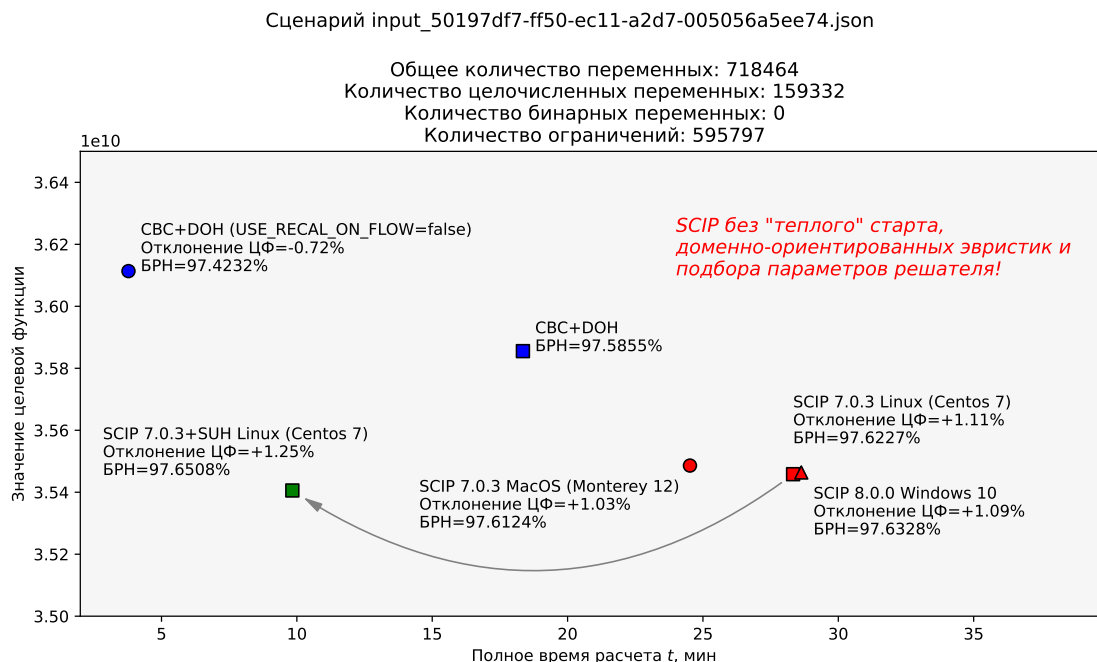


Рис. 15. Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий 50197df7 без бинарных переменных

### 11.1.3. Сценарий 7FAC4231 без бинарных переменных

#### Статистика

Общее количество переменных: 737585

Количество целочисленных переменных: 147789

Количество бинарных переменных: 0

Количество ограничений: 540018

lp-файл: <https://disk.yandex.ru/d/qiZAmraUNK1Peg>

#### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- о INTSHIFING,
- о RENS.

Файл решения задачи (метаконфигурация SUH) доступен по ссылке [https://disk.yandex.ru/d/20NeMuQ7NF\\_ccA](https://disk.yandex.ru/d/20NeMuQ7NF_ccA)

Файл статистической сводки (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/QxE0HoREHzgHQQ>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке [https://disk.yandex.ru/d/FHZGj\\_Kyg8dDiw](https://disk.yandex.ru/d/FHZGj_Kyg8dDiw)

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/8H1vw6zkQS7DAg>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (`USE_RECALCULATION_ON_FLOW=true`) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 5.22% лучше в смысле целевой функции и на 27.10% – в смысле временных издержек (рис. 16).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 5.452% лучше в смысле целевой функции и на 90.16% – в смысле временных издержек (табл. 3).

Синим цветом обозначен выигрыш в процентах.

Таблица 3. Сводка результатов анализа эффективности метаконфигураций SUH и FZBIVSUHPB. Сценарий 7fac4231 без бинарных переменных

Способ	Полное время расчета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	16.05	1.087609
SCIP+SUH	11.67 +27.29%	1.030866 +5.222%
SCIP+FZB...	3.58 +77.69%	1.028349 +5.452%

### 11.1.4. Сценарий SA485A55 без бинарных переменных

#### Статистика

Общее количество переменных: 718601

Количество целочисленных переменных: 140858

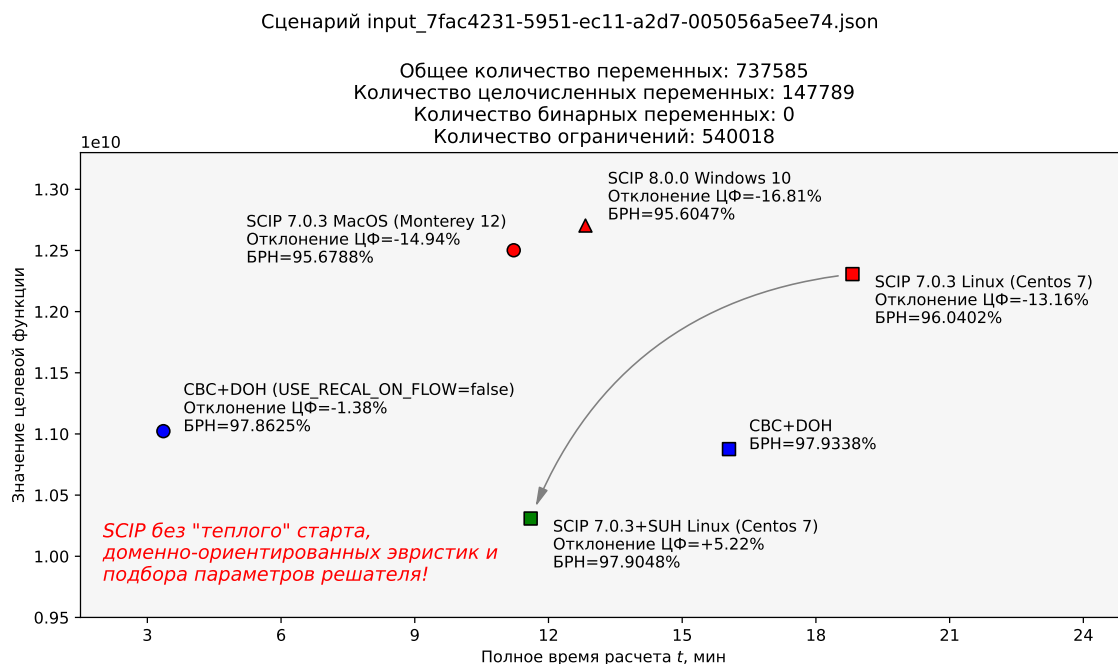


Рис. 16. Сводка результатов анализа эффективности метаконфигурации SUH.  
Сценарий 7fac4231 без бинарных переменных

Количество бинарных переменных: 0

Количество ограничений: 514229

лр-файл: [https://disk.yandex.ru/d/iSP6xrh4K\\_wHEQ](https://disk.yandex.ru/d/iSP6xrh4K_wHEQ)

### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

Файл решения задачи (метаконфигурация SUH) доступен по ссылке [https://disk.yandex.ru/d/\\_WzkmgoueNb2Bg](https://disk.yandex.ru/d/_WzkmgoueNb2Bg)

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/sLUW5IxmPMBpcw>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/3Ls6QrAWVUMdZw>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 0.683% лучше в смысле целевой функции и на 46.48% – в смысле временных издержек (рис. 17).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 1.244% лучше в смысле целевой функции и на 88.53% – в смысле временных издержек (табл. 4).

Синим цветом обозначен выигрыш в процентах.

Таблица 4. Сводка результатов анализа эффективности метаконфигураций SUN и FZBIVSUHPB. Сценарий **ca485a55** без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	20.05	4.597048
SCIP+SUN	10.73 +46.48%	4.565579 +0.683%
SCIP+FZB...	4.34 +78.35%	4.539819 +1.244%

Сценарий input\_ca485a55-0485-ec11-a2db-005056a5ee74.json

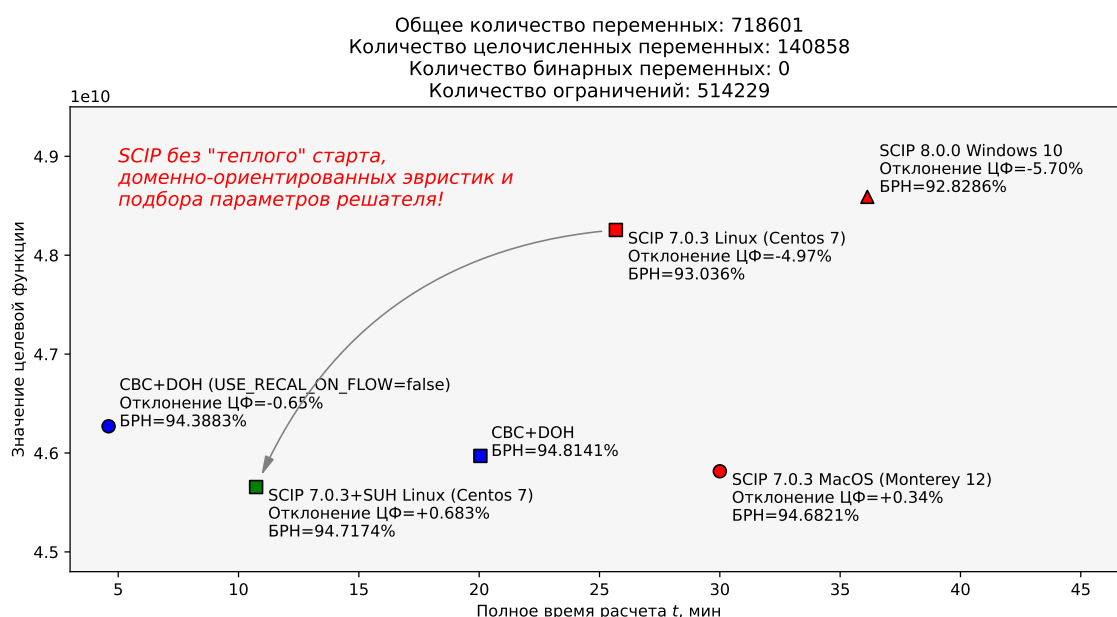


Рис. 17. Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий **ca485a55** без бинарных переменных

#### 11.1.5. Сценарий 276 без бинарных переменных

##### Статистика

Общее количество переменных: 809224

Количество целочисленных переменных: 162562

Количество бинарных переменных: 0

Количество ограничений: 602190

lp-файл: <https://disk.yandex.ru/d/QaS5kd7VRZQ66A>

##### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

Файл решения задачи (метаконфигурация SUN) доступен по ссылке <https://disk.yandex.ru/d/M2V88djiiGM5PA>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/G0ustAVT6I9CeA>



Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/YBXB5GCECJiBIA>

**Вывод по сценарию:** описанная выше метаконфигурация SUN приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 3.67% лучше в смысле целевой функции и на 51.56% – в смысле временных издержек (рис. 18).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 4.86% лучше в смысле целевой функции и на 78.35% – в смысле временных издержек (табл. 5).

Синим цветом обозначен выигрыш в процентах.

Таблица 5. Сводка результатов анализа эффективности метаконфигураций SUN и FZBIVSUHPB. Сценарий 276 без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	29.87	1.430789
SCIP+SUN	14.47 +51.56%	1.378299 +3.669%
SCIP+FZB...	3.95 +78.35%	1.361368 +4.857%

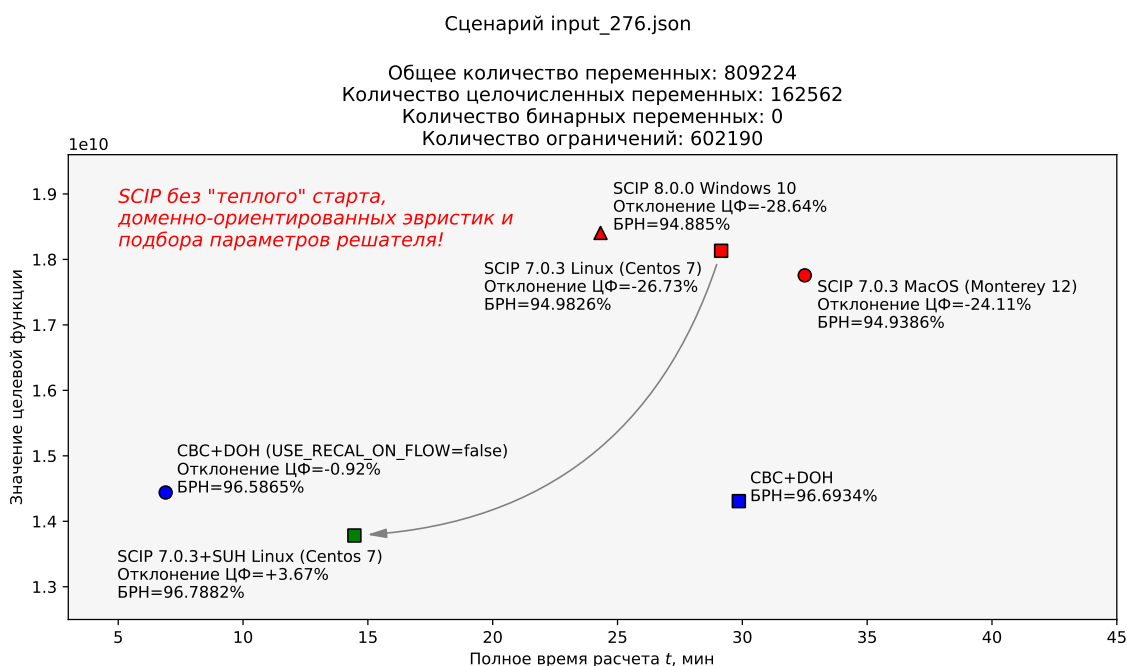


Рис. 18. Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий 276 без бинарных переменных

#### 11.1.6. Сценарий 337 без бинарных переменных

##### Статистика

Общее количество переменных: 859075

Количество целочисленных переменных: 173622

Количество бинарных переменных: 0

Количество ограничений: 624327

лр-файл: <https://disk.yandex.ru/d/keyQLAagsD7Sbw>

### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

Файл решения задачи (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/ZUIEo3dDq77FjA>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/OnUXIrIKuzqZlw>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/UONCnMQN1akHUA>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 22.12% лучше в смысле целевой функции и на 18.32% – в смысле временных издержек (рис. 19).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 22.59% лучше в смысле целевой функции и на 70.84% – в смысле временных издержек (табл. 6).

Синим цветом обозначен выигрыш в процентах.

Таблица 6. Сводка результатов анализа эффективности метаконфигураций SUH и FZBIVSUHPB. Сценарий 337 без бинарных переменных

Способ	Полное время расчета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	20.85	3.825042
SCIP+SUH	17.03 +18.32%	2.978782 +22.123%
SCIP+FZB...	6.08 +70.84%	2.961019 +22.588%

#### 11.1.7. Сценарий 13D686AB без бинарных переменных

##### Статистика

Общее количество переменных: 786020

Количество целочисленных переменных: 168857

Количество бинарных переменных: 0

Количество ограничений: 598414

лр-файл: <https://disk.yandex.ru/d/3KkYKzNl3PjGdg>

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

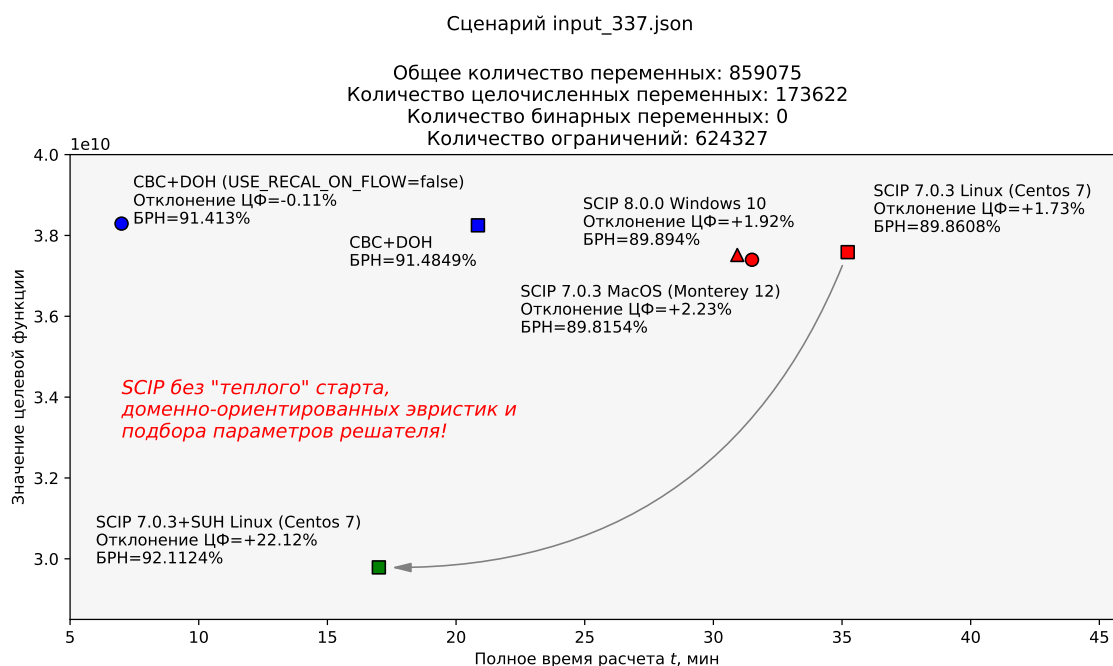


Рис. 19. Сводка результатов анализа эффективности метаконфигурации SUH.  
Сценарий 337 без бинарных переменных

Файл решения задачи (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/EXylMeX6Ytz4tg>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/dXUMVbSWRbqeDQ>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/Knavj89muxGw-w>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках (`USE_RECALCULATION_ON_FLOW=true`) для последнего решения из пула допустимых целочисленных решений (OC Linux Centos 7) на 9.40% лучше в смысле целевой функции и на 33.03% – в смысле временных издержек (рис. 20).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к тому же результату на доменно-ориентированных эвристиках дает решение задачи, которое на 10.44% лучше в смысле целевой функции и на 75.82% – в смысле временных издержек (табл. 7).

Синим цветом обозначен выигрыш в процентах.

Таблица 7. Сводка результатов анализа эффективности метаконфигураций SUH и FZBIVSUHPB. Сценарий 13d686ab без бинарных переменных

Способ	Полное время расчета, мин	Верхняя граница решения, $\times 10^9$
CBC+DOH	28.82	8.774743
SCIP+SUH	19.30 +33.03%	7.949568 +9.403%
SCIP+FZB...	6.97 +75.82%	7.858548 +10.441%

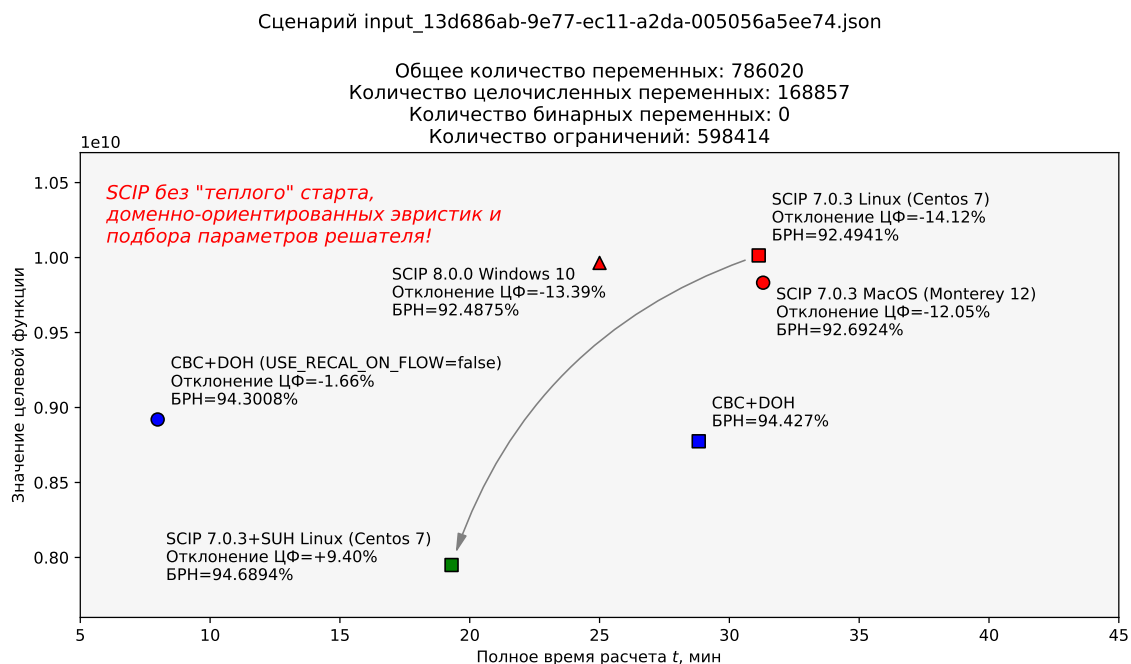


Рис. 20. Сводка результатов анализа эффективности метаконфигурации SUH.  
Сценарий 13d686ab без бинарных переменных

#### 11.1.8. Сценарий A78CBEAD без бинарных переменных

##### Статистика

Общее количество переменных: 795400

Количество целочисленных переменных: 180160

Количество бинарных переменных: 0

Количество ограничений: 658339

lp-файл: <https://disk.yandex.ru/d/vTPPa1H3VFD7tA>

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- RENS.

Файл решения задачи (метаконфигурация SUH) доступен по ссылке <https://disk.yandex.ru/d/fARVcHb66ToHxQ>

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/4DIteZTja77cog>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/vn1K834mY5MEng>

**Вывод по сценарию:** описанная выше метаконфигурация SUH приводит к решению задачи, которое оказывается по отношению к приему на доменно-ориентированных эвристиках (USE\_RECALCULATION\_ON\_FLOW=true) для последнего решения из пула допустимых целочисленных решений (ОС Linux Centos 7) на 1.57% лучше в смысле целевой функции и на 23.30% – в смысле временных издержек (рис. 21).

Метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к приему построения решения на доменно-ориентированных эвристиках дает решение задачи, которое на 1.39% лучше в смысле целевой функции и на 81.04% – в смысле временных издержек (табл. 8).

Синим цветом обозначен выигрыш в процентах.

Таблица 8. Сводка результатов анализа эффективности метаконфигураций SUN и FZBIVSUNPB. Сценарий a78cbead без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	26.05	3.801546
SCIP+SUN	19.98 +23.30%	3.741685 +1.576%
SCIP+FZB...	4.94 +81.04%	3.748890 +1.386%

Сценарий input\_a78cbead-073b-ec11-a2d4-005056a5ee74.json

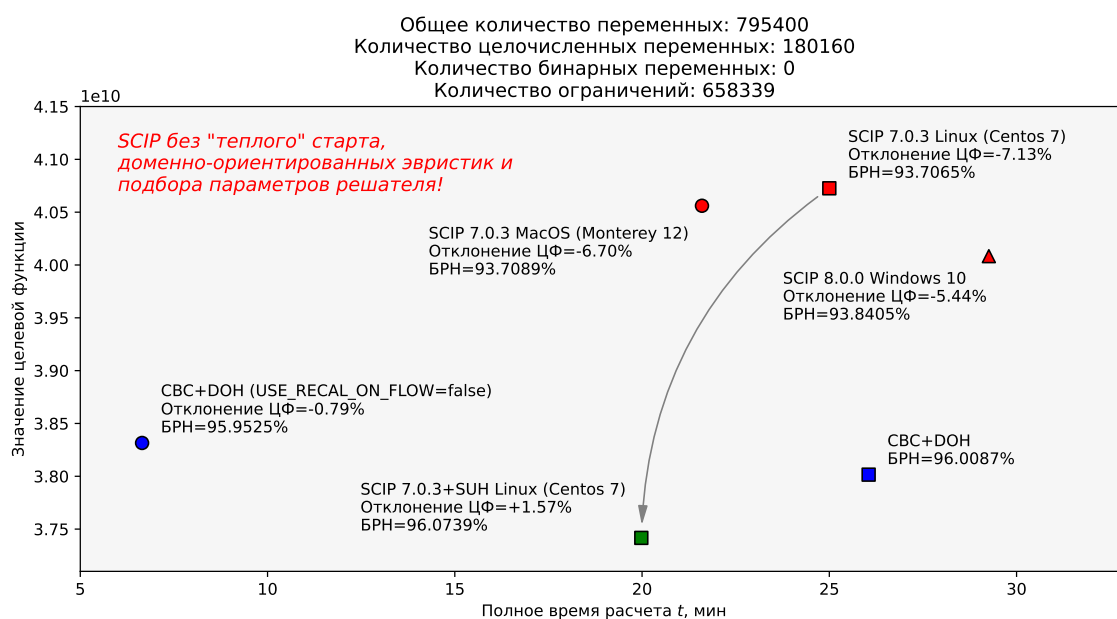


Рис. 21. Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий a78cbead без бинарных переменных

#### 11.1.9. Сценарий 496 (hard) без бинарных переменных

##### Статистика<sup>23</sup>

Общее количество переменных: 864743 (48862) (90762)

Количество целочисленных переменных: 177365 (5008) (25872)

Количество бинарных переменных: 0 (332) (27)

Количество ограничений: 610819 (25438) (39119)

lp-файл: <https://disk.yandex.ru/d/CUA7wSn35k7Gbw>

Решение задачи было найдено с помощью первичной эвристики INTSHIFTING.

Файл решения задачи (метаконфигурация FZBIVSUNPB) доступен по ссылке <https://disk.yandex.ru/d/tbMiAbYmaA0rhg>

Файл статистической сводки (метаконфигурация FZBIVSUNPB) доступен по ссылке <https://disk.yandex.ru/d/AQptE3s3NF4bug>

<sup>23</sup> В первых скобках указана размерность задачи после шага пресолвинга с фиксацией FZBIVSUNPB, а во вторых – с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов

Файл решения задачи (ансамбль детекторов аномалий) доступен по ссылке <https://disk.yandex.ru/d/VMZLFWoT80ftXA>

Файл статистической сводки (ансамбль детекторов аномалий) доступен по ссылке <https://disk.yandex.ru/d/KckqXgoKfv2fyQ>

Решение SCIP+ML получено с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов.

**Вывод по сценарию:** метаконфигурация FZBIVSUNPB (подробнее в разделе 11.2) по отношению к приему на доменно-ориентированных эвристиках CBC+MS (measure of similarity) дает решение задачи, которое на 9.823% лучше в смысле целевой функции и на 69.13% – в смысле временных издержек (табл. 9).

Решение, полученное с помощью ансамбля детекторов аномалий, обученного на сценарии f398266b\_bin.lp, на 9.678% превосходит CBC+MS в смысле целевой функции и на 71.82% – в смысле временных издержек.

SCIP+ML(0.10)f – решение, полученное с помощью ансамбля детекторов аномалий без подбора параметра контаминации при первом запуске приложения, SCIP+ML(0.10)e – то же самое, при запуске приложения в «исследовательском режиме» (матрица ограничений обучающего поднабора данных и релаксированные решения не вычисляются повторно).

Синим цветом обозначен выигрыш в процентах.

Таблица 9. Сводка результатов анализа эффективности метаконфигураций FZBIVSUNPB и ансамбля детекторов аномалий.  
Сценарий 496 без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^7$
CBC+MS*	5.00	6.536728
Gurobi 9.12	5.22 <span style="color: red;">-0.04%</span>	5.834197 <span style="color: blue;">+10.747%</span>
SCIP 7.0.3d**	15.42 <span style="color: red;">-66.15%</span>	10.66377 <span style="color: red;">-38.702%</span>
SCIP+FZB...	1.54 <span style="color: blue;">+69.13%</span>	5.894658 <span style="color: blue;">+9.823%</span>
SCIP+ML(0.10)f	4.56 <span style="color: blue;">+8.8%</span>	5.904120 <span style="color: blue;">+9.678%</span>
SCIP+ML(0.10)e	1.51 <span style="color: blue;">+69.76%</span>	5.904120 <span style="color: blue;">+9.678%</span>

\* – опорное решение

\*\* – решение было прервано

#### 11.1.10. Сценарий 514 (hard) без бинарных переменных

##### Статистика<sup>24</sup>

Общее количество переменных: 775879 (77367) (120764)

Количество целочисленных переменных: 145292 (5817) (32895)

Количество бинарных переменных: 0 (30) (14)

Количество ограничений: 541040 (45892) (61074)

lp-файл: <https://disk.yandex.ru/d/jQqSqBKb6iG-vw>

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFTING,
- RENS.

<sup>24</sup>В первых скобках указана размерность задачи после шага пресолвинга с фиксацией FZBIVSUNPB, а во вторых – с фиксацией, полученной с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/1N2FdsqwEQcVTQ>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/iIdbACgh59EpVg>

Файл решения задачи (ансамбль детекторов аномалий) доступен по ссылке <https://disk.yandex.ru/d/5kRy0UsI0atHsQ>

Файл статистической сводки (ансамбль детекторов аномалий) доступен по ссылке <https://disk.yandex.ru/d/rNUU8HmeBGLFRQ>

Решение SCIP+ML получено с помощью ансамбля детекторов аномалий без подбора гипер-параметров детекторов.

**Вывод по сценарию:** метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к приему построения решения с помощью меры подобия CBC+MS (measure of similarity) дает решение задачи, которое на 18.616% лучше в смысле целевой функции и на 51.82% хуже в смысле временных издержек (табл. 10).

Решение, полученное с помощью ансамбля детекторов аномалий<sup>25</sup>, обученного на сценарии f398266b\_bin.lp, на 19.562% превосходит CBC+MS в смысле целевой функции и на 6.31% – в смысле временных издержек.

SCIP+ML(0.10)f – решение, полученное с помощью ансамбля детекторов аномалий без подбора параметра контаминации при первом запуске приложения, SCIP+ML(0.10)e – то же самое, при запуске приложения в «исследовательском режиме» (матрица ограничений обучающего под-набора данных и релаксированные решения не вычисляются повторно).

Синим цветом обозначен выигрыш в процентах, а красным – проигрыш.

Таблица 10. Сводка результатов анализа эффективности метаконфигураций FZBIVSUHPB и ансамбля детекторов аномалий.  
Сценарий 514 без бинарных переменных

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^9$
CBC+MS*	13.00	5.243829
Gurobi 9.12	11.(6) +10.31	4.239092 +19.160%
SCIP 7.0.3d**	60.32 -79.47%	47.82659 -89.036%
SCIP+FZB...	26.98 -51.82%	4.267692 +18.616%
SCIP+ML(0.10)f	12.171 +6.38%	4.217134 +19.580%
SCIP+ML(0.10)e	6.53 +49.77%	4.217134 +19.580%

\* – опорное решение

\*\* – решение было прервано

#### 11.1.11. Сценарий 519 (hard) без бинарных переменных

##### Статистика<sup>26</sup>

Общее количество переменных: 684412 (75034)

Количество целочисленных переменных: 159200 (5424)

Количество бинарных переменных: 0 (44)

Количество ограничений: 447182 (44735)

lp-файл: <https://disk.yandex.ru/d/MMvnnYXK4J4Xxw>

<sup>25</sup>Решение принудительно останавливалось на 350 секунде (параметр limits/softtime = 350)

<sup>26</sup>В скобках указана размерность задачи после шага пресолвинга с фиксацией FZBIVSUHPB



Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFING,
- ONEOPT,
- VECLENDI,
- LINESEARCH,
- RENS.

Файл решения задачи (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/25B3mUiRYdid3A>

Файл статистической сводки (метаконфигурация FZBIVSUHPB) доступен по ссылке <https://disk.yandex.ru/d/L3TyaXp56rZjCA>

Файл решения задачи (ансамбль детекторов аномалий) доступен по ссылке

Файл статистической сводки (ансамбль детекторов аномалий) доступен по ссылке

Решение SCIP+ML получено с помощью ансамбля детекторов аномалий без подбора гиперпараметров детекторов.

**Вывод по сценарию:** метаконфигурация FZBIVSUHPB (подробнее в разделе 11.2) по отношению к приему построения решения на доменно-ориентированных эвристиках CBC+MS (measure of similarity) дает решение задачи, которое на % лучше в смысле целевой функции и на % хуже в смысле временных издержек (табл. 11).

Решение, полученное с помощью отдельного детектора аномалий, обученного на сценарии f398266b\_bin.lp, на % превосходит CBC+MS в смысле целевой функции и на % – в смысле временных издержек.

SCIP+ML(0.10)f – решение, полученное с помощью ансамбля детекторов аномалий без подбора параметра контаминации при первом запуске приложения, SCIP+ML(0.10)e – то же самое, при запуске приложения в «исследовательском режиме» (матрица ограничений обучающего поднабора данных и релаксированные решения не вычисляются повторно).

Синим цветом обозначен выигрыш в процентах, а красным – проигрыш.

Таблица 11. Сводка результатов анализа эффективности метаконфигураций FZBIVSUHPB и ансамбля детекторов аномалий. Сценарий 519 без бинарных переменных

Способ	Полное время расчета, мин	Верхняя граница решения, $\times 10^7$
CBC+MS*	6.00	7.719212
Gurobi 9.12	3.48 +42.00%	7.062839 +8.503%
SCIP 7.0.3d**	41.92 -91.70%	31.59748 +77.647%
SCIP+FZB (a)	5.23 +12.83%	7.901148 -2.302%
SCIP+FZB (b)	28.83 -79.19%	7.374810 +4.462%
SCIP+ML(0.10)f	42.07 -85.74 %	7.014369 +9.130%

\* – опорное решение

\*\* – решение было прервано

## 11.2. Поиск решения на сценариях с бинарными переменными.

### Метаконфигурация FZBIVSUHPB

На ранних стадиях изучения проблемы высокоразмерных сценариев с бинарными переменными, поиск решения осуществлялся в семь шагов:

1. Подавить подгруппу первичных эвристик низкой эффективности (см. раздел 9.3),



2. При разрешении конфликтов и ветвлении<sup>27</sup> отдавать предпочтение бинарным переменным,
3. Найти релаксированное решение задачи,
4. Подобрать порог бинаризации на релаксированном решении для бинарных переменных (см. раздел 9.4),
5. Зафиксировать *нулевые 0-bin и единичные 1-bin бинарные переменные*; подать фиксацию решателю,
6. В решении, найденном на предыдущей итерации, зафиксировать *нулевые целочисленные 0-int и единичные бинарные 1-bin переменные*; полученную фиксацию подать на вход решателя,
7. В решении, полученном на предыдущей итерации, зафиксировать *нулевые бинарные 0-bin и целочисленные 0-int переменные*; фиксацию подать на вход решателя.

Процедура поиска оказалась чувствительной к параметру `autorestartnodes`. Графическая интерпретация результатов вычислительных экспериментов с разверткой процедуры поиска верхней границы решения во времени приведена на рис. 22, 23, 24 и 25.

Позже описанную процедуру удалось упростить и свести к следующей *метаконфигурации* FZBIVSUNPB (Fixed Zero Binary and Integer Variables, Suppress Useless Heuristics, Prefer Binary):

1. Подавить подгруппу первичных эвристик низкой эффективности,
2. При разрешении конфликтов и ветвлении отдавать предпочтение *бинарным* переменным,
3. Зафиксировать *нулевые бинарные 0-bin и нулевые целочисленные 0-int переменные* в релаксированном решении (см. раздел 9.1).

Конфигурация решателя SCIP для всех сценариев группы ИКП (с бинарными переменными) имеет вид

scip.set. Сценарии группы ИКП с бинарными переменными

```
# критерии останова и перезапуска
limits/time = 7200
limits/autorestartnodes = -1
limits/gap = 0.02 # решение останавливается при зазоре <= 2%

# управление стратегиями анализа конфликтов и ветвления
conflict/preferbinary = True
branching/preferbinary = True

# подавление подгруппы первичных эвристик низкой эффективности
heuristics/farkasdiving/freq = -1
heuristics/feaspump/freq = -1
heuristics/randrounding/freq = -1
heuristics/shiftandpropagate/freq = -1
heuristics/shifting/freq = -1
```

Все эксперименты проводились на виртуальной машине Linux (Centos 7) Intel Core™ i7 (8 CPUs), 3.6GHz, RAM 16Gb.

Сводка результатов вычислительных экспериментов доступна по ссылке <https://docs.google.com/document/d/1V9fZLT9cXkbVQ5BvMCwzKrAiASZ2v4-01Z68jVBZUBU/edit?usp=sharing>.

Кодовая база решения доступна по ссылке [https://gitdp.zyfra.com/ds\\_and\\_math\\_users/ml-dl-in-operations-reaseearches.git](https://gitdp.zyfra.com/ds_and_math_users/ml-dl-in-operations-reaseearches.git)

<sup>27</sup>К сожалению, на сценариях группы ИКП с бинарными переменными решателю SCIP не удастся найти решение в корне дерева

### 11.2.1. Сценарий A78CBEAD с бинарными переменными

#### Статистика

Общее количество переменных: 797818

Количество целочисленных переменных: 180160

Количество бинарных переменных: 2418

Количество ограничений: 663175

lp-файл: <https://disk.yandex.ru/d/JbT3KR5Yi1ZomQ>

#### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- DISTRIBUTIOINDIVING,
- ONEOPT,
- GINS.

Фрагмент лога сессии SCIP

```
...
time | node | left | LP iter|LP it/n|mem/heur|mdpt |vars |cons |rows |cuts |sepa|confs|strbr|
      dualbound | primalbound | gap | compl.
d1790s| 1881 | 1668 | 1010k| 296.9 |distrib| 93 | 50k| 43k| 43k| 0 | 1 | 385 |3585 |
      3.757279e+10 | 3.894342e+10 | 3.65%| 7.70%
d1790s| 1881 | 1668 | 1010k| 296.9 |distrib| 93 | 50k| 43k| 43k| 0 | 1 | 385 |3585 |
      3.757279e+10 | 3.894341e+10 | 3.65%| 7.70%
i1792s| 1882 | 1667 | 1011k| 297.0 | oneopt| 93 | 50k| 43k| 43k|8612 | 0 | 385 |3585 |
      3.757279e+10 | 3.893993e+10 | 3.64%| 7.70%
1796s| 1900 | 1687 | 1016k| 297.0 | 3669M | 93 | 50k| 43k| 43k|8644 | 1 | 387 |3585 |
      3.757279e+10 | 3.893993e+10 | 3.64%| 2.82%
L1902s| 1982 | 1769 | 1090k| 313.4 | gins| 93 | 50k| 43k| 43k|8935 | 1 | 398 |3590 |
      3.757279e+10 | 3.875897e+10 | 3.16%| 2.83%
L1912s| 1982 | 1769 | 1090k| 313.4 | gins| 93 | 50k| 43k| 43k|8935 | 1 | 398 |3590 |
      3.757279e+10 | 3.864257e+10 | 2.85%| 2.83%
i1920s| 1982 | 1769 | 1099k| 316.2 | oneopt| 93 | 50k| 43k| 43k|8935 | 1 | 398 |3590 |
      3.757279e+10 | 3.864241e+10 | 2.85%| 2.83%
1954s| 2000 | 1787 | 1133k| 325.5 | 3731M | 93 | 50k| 43k| 43k|9004 | 1 | 398 |3591 |
      3.757279e+10 | 3.864241e+10 | 2.85%| 2.83%
```

Файл решения задачи доступен по ссылке <https://disk.yandex.ru/d/6FPE-S5VupA6iw>

Файл статистической сводки доступен по ссылке <https://disk.yandex.ru/d/9G-v54ywEK1TJA>

**Вывод по сценарию:** описанная выше метаконфигурация приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках для последнего решения из пула допустимых целочисленных решений на 2.46% лучше в смысле целевой функции и на 19.64% – в смысле временных издержек (табл. 12).

В табл. 12 через SCIP+MC (a) обозначается решение, построенное на метаконфигурации SCIP, отвечающее *первому* допустимому целочисленному решению, верхняя граница которого не превышает верхнюю границу решения на доменно-ориентированных эвристиках, а через SCIP+MC (b) – решение, отвечающее *последнему* допустимому целочисленному решению в наборе полученных.

Синим цветом обозначен выигрыш в процентах.

Таблица 12. Сводка результатов анализа эффективности метаконфигурации FZBIVSUHPB. Сценарий a78cbead с бинарными переменными

Способ	Полное время рас- чета, мин		Верхняя граница решения, $\times 10^{10}$	
CBC+DOH	39.82		3.961502	
SCIP+MC (a)	29.83	+25.09%	3.894342	+1.70%
SCIP+MC (b)	32.00	+19.64%	3.864241	+2.46%

### 11.2.2. Сценарий 7FAC4231 с бинарными переменными

#### Статистика

Общее количество переменных: 740251

Количество целочисленных переменных: 147789

Количество бинарных переменных: 2666

Количество ограничений: 545350

lp-файл: <https://disk.yandex.ru/d/3NbbjflW5zhejq>

#### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFTING,
- ONEOPT,
- GINS,
- CROSSOVER,
- ALNS.

Фрагмент лога сессии SCIP

```
...
time | node | left | LP iter|LP it/n|mem/heur|mdpt |vars |cons |rows |cuts |sepa|confs|strbr|
dualbound | primalbound | gap | compl.
r 454s| 372 | 341 | 91171 | 102.3 |intshift| 309 | 41k| 33k| 34k|2788 | 5 | 57 |3711 |
1.053077e+10 | 1.309195e+10 | 24.32%| 0.78%
i 454s| 373 | 340 | 91171 | 102.0 | oneopt| 309 | 41k| 33k| 34k|2788 | 0 | 57 |3711 |
1.053077e+10 | 1.308634e+10 | 24.27%| 0.78%
463s| 400 | 369 | 93623 | 101.3 | 2493M | 309 | 41k| 33k| 34k|2950 | 1 | 57 |3761 |
1.053077e+10 | 1.308634e+10 | 24.27%| 0.29%
L 507s| 473 | 442 |106991 | 113.9 | gins| 309 | 41k| 33k| 34k|3084 | 1 | 57 |3813 |
1.053077e+10 | 1.297515e+10 | 23.21%| 0.29%
L 512s| 473 | 442 |106991 | 113.9 | gins| 309 | 41k| 33k| 34k|3084 | 1 | 57 |3813 |
1.053077e+10 | 1.292548e+10 | 22.74%| 0.29%
L 522s| 473 | 442 |106991 | 113.9 | gins| 309 | 41k| 33k| 34k|3084 | 1 | 57 |3813 |
1.053077e+10 | 1.289283e+10 | 22.43%| 0.29%
L 525s| 473 | 442 |106991 | 113.9 | gins| 309 | 41k| 33k| 34k|3084 | 1 | 57 |3813 |
1.053077e+10 | 1.286340e+10 | 22.15%| 0.29%
i 529s| 473 | 442 |112279 | 125.1 | oneopt| 309 | 41k| 33k| 34k|3084 | 1 | 57 |3813 |
1.053077e+10 | 1.285668e+10 | 22.09%| 0.29%
r 531s| 474 | 443 |120630 | 142.5 |intshift| 309 | 41k| 33k| 34k|3084 | 1 | 58 |3813 |
1.053077e+10 | 1.197786e+10 | 13.74%| 0.29%
i 532s| 474 | 373 |124926 | 151.6 | oneopt| 309 | 41k| 33k| 34k|3084 | 1 | 58 |3813 |
1.053077e+10 | 1.197230e+10 | 13.69%| 0.29%
536s| 500 | 399 |126496 | 146.9 | 2579M | 309 | 41k| 33k| 34k|3181 | 1 | 58 |3822 |
1.053077e+10 | 1.197230e+10 | 13.69%| 0.29%
567s| 600 | 499 |158520 | 175.8 | 2613M | 309 | 41k| 33k| 34k|3641 | 1 | 60 |3933 |
1.053095e+10 | 1.197230e+10 | 13.69%| 0.29%
```

L 739s	659	554  189783	207.6	gins	309	41k	33k	34k 4060	1	62  3978
	1.053095e+10	1.191898e+10	13.18%	0.29%						
i 741s	660	555  198453	220.4	oneopt	309	41k	33k	34k 4060	1	62  3981
	1.053095e+10	1.191889e+10	13.18%	0.30%						
794s	700	595  236166	261.7	2689M	309	41k	33k	34k 4418	1	62  4010
	1.053095e+10	1.191889e+10	13.18%	0.32%						
836s	800	695  277232	280.4	2728M	309	41k	33k	34k 4757	1	64  4027
	1.053219e+10	1.191889e+10	13.17%	0.32%						
L 967s	860	693  295017	281.5	crossove	309	41k	33k	34k 5000	1	64  4059
	1.053219e+10	1.154287e+10	9.60%	0.32%						
i 968s	860	693  300734	288.1	oneopt	309	41k	33k	34k 5000	1	64  4059
	1.053219e+10	1.154284e+10	9.60%	0.32%						
990s	900	733  312921	288.9	2793M	309	41k	33k	34k 5288	1	64  4139
	1.053219e+10	1.154284e+10	9.60%	0.33%						
1042s	1000	823  346085	293.2	2816M	309	41k	33k	34k 5725	1	65  4281
	1.053219e+10	1.154284e+10	9.60%	0.33%						
L1083s	1003	826  347173	293.4	alns	309	41k	33k	34k 5747	2	65  4284
	1.053219e+10	1.153273e+10	9.50%	0.33%						
i1084s	1004	827  352908	298.8	oneopt	309	41k	33k	34k 5747	1	65  4284
	1.053219e+10	1.118743e+10	6.22%	0.33%						
1113s	1100	699  373504	291.4	2860M	309	41k	33k	34k 6055	3	65  4323
	1.053219e+10	1.118743e+10	6.22%	0.44%						
1140s	1	0  419115	-	3039M	0	41k	34k	34k  0   0	65  4323	
	1.053219e+10	1.118743e+10	6.22%	unknown						

Файл решения задачи доступен по ссылке <https://disk.yandex.ru/d/TmA6hqFV87eGTg>

Файл статистической сводки доступен по ссылке [https://disk.yandex.ru/d/CsGV\\_oal40Tx0Q](https://disk.yandex.ru/d/CsGV_oal40Tx0Q)

**Вывод по сценарию:** описанная выше метаконфигурация приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках для последнего решения из пула допустимых целочисленных решений на 3.38% лучше в смысле целевой функции и на 33.07% – в смысле временных издержек (табл. 13).

В табл. 13 через SCIP+MC (a) обозначается решение, построенное на метаконфигурации SCIP, отвечающее *первому* допустимому целочисленному решению, верхняя граница которого не превышает верхнюю границу решения на доменно-ориентированных эвристиках, а через SCIP+MC (b) – решение, отвечающее *последнему* допустимому целочисленному решению в наборе полученных.

Синим цветом обозначен выигрыш в процентах.

Таблица 13. Сводка результатов анализа эффективности метаконфигурации FZBIVSUHPB. Сценарий 7fac4231 с бинарными переменными

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	27.00	1.157865
SCIP+MC (a)	18.05 <span style="color: blue;">+33.15%</span>	1.153273 <span style="color: blue;">+0.40%</span>
SCIP+MC (b)	18.07 <span style="color: blue;">+33.07%</span>	1.118743 <span style="color: blue;">+3.38%</span>

### 11.2.3. Сценарий 50197DF7 с бинарными переменными

#### Статистика

Общее количество переменных: 720954

Количество целочисленных переменных: 159332

Количество бинарных переменных: 2490

Количество ограничений: 600777

lp-файл: <https://disk.yandex.ru/d/qWeSKb2WEs6kQA>

### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFTING,
- ONEOPT,
- GINS.

Фрагмент лога сессии SCIP

...														
time	node	left	LP iter	LP it/n	mem/heur	mdpt	vars	cons	rows	cuts	sepa	confs	strbr	
dualbound			primalbound		gap		compl.							
r 836s	963	948	155676	53.5	intshift	409	41k	34k	35k	4367	1	69	7354	
3.554610e+10			3.676991e+10		3.44%		unknown							
i 836s	964	947	155676	53.5	oneopt	409	41k	34k	35k	4367	0	69	7354	
3.554610e+10			3.676497e+10		3.43%		unknown							
846s	1000	985	157559	53.4	2577M	409	41k	34k	35k	4396	1	69	7444	
3.554610e+10			3.676497e+10		3.43%		unknown							
L 885s	1064	1049	157869	50.5	gins	409	41k	34k	35k	4397	1	69	7484	
3.554610e+10			3.659894e+10		2.96%		unknown							
L 931s	1064	1049	157869	50.5	gins	409	41k	34k	35k	4397	1	69	7484	
3.554610e+10			3.656967e+10		2.88%		unknown							
i 962s	1064	1049	161589	54.0	oneopt	409	41k	34k	35k	4397	1	69	7484	
3.554610e+10			3.656967e+10		2.88%		unknown							
969s	1100	1085	161769	52.4	2620M	409	41k	34k	35k	4397	1	69	7532	
3.554610e+10			3.656967e+10		2.88%		unknown							
L 988s	1164	1149	161992	49.7	gins	409	41k	34k	35k	4397	1	69	7557	
3.554610e+10			3.630031e+10		2.12%		unknown							
L 993s	1164	1149	161992	49.7	gins	409	41k	34k	35k	4397	1	69	7557	
3.554610e+10			3.625804e+10		2.00%		unknown							
L1000s	1164	1149	161992	49.7	gins	409	41k	34k	35k	4397	1	69	7557	
3.554610e+10			3.623675e+10		1.94%		unknown							

Файл решения задачи доступен по ссылке [https://disk.yandex.ru/d/2\\_FDqS70q0UBqA](https://disk.yandex.ru/d/2_FDqS70q0UBqA)

Файл статистической сводки доступен по ссылке <https://disk.yandex.ru/d/SkRLorYzQDI-Aw>

**Вывод по сценарию:** описанная выше метаконфигурация приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках для последнего решения из пула допустимых целочисленных решений на 2.87% лучше в смысле целевой функции и на 36.08% – в смысле временных издержек (табл. 14).

В табл. 14 через SCIP+MC (a) обозначается решение, построенное на метаконфигурации SCIP, отвечающее *первому* допустимому целочисленному решению, верхняя граница которого не превышает верхнюю границу решения на доменно-ориентированных эвристиках, а через SCIP+MC (b) – решение, отвечающее *последнему* допустимому целочисленному решению в наборе полученных.

Синим цветом обозначен выигрыш в процентах.

#### 11.2.4. Сценарий F398266B с бинарными переменными

##### Статистика

Общее количество переменных: 777271

Таблица 14. Сводка результатов анализа эффективности метаконфигурации FZBIVSUHPB. Сценарий 50197df7 с бинарными переменными

Способ	Полное время рас- чета, мин		Верхняя граница решения, $\times 10^{10}$	
CBC+DOH	28.27		3.730552	
SCIP+MC (a)	13.93	+50.73%	3.676991	+1.44%
SCIP+MC (b)	18.07	+36.08%	3.623675	+2.87%

Количество целочисленных переменных: 172449

Количество бинарных переменных: 2370

Количество ограничений: 655003

lp-файл: <https://disk.yandex.ru/d/4YFYJSB1I1wsmQ>

### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- DISTRIBUTIOINDIVING,
- ONEOPT,
- CROSSOVER.

Фрагмент лога сессии SCIP

...														
time	node	left	LP iter	LP it/n	mem/heur	mdpt	vars	cons	rows	cuts	sepa	confs	strbr	
dualbound			primalbound	gap	compl.									
d1163s	433	434	462507	790.8	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054807e+10	3.36%	unknown									
d1164s	433	434	462644	791.1	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054779e+10	3.36%	unknown									
d1164s	433	434	462746	791.3	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054778e+10	3.36%	unknown									
d1164s	433	434	462780	791.4	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054776e+10	3.36%	unknown									
d1164s	433	434	462801	791.4	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054776e+10	3.36%	unknown									
d1165s	433	434	462836	791.5	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054776e+10	3.36%	unknown									
d1165s	433	434	462856	791.6	distribu	51	59k	48k	49k	0	1	17	1387	
	5.857793e+10		6.054774e+10	3.36%	unknown									
i1167s	434	433	463020	790.1	oneopt	51	59k	48k	49k	4333	0	17	1387	
	5.857793e+10		6.053918e+10	3.35%	unknown									
1250s	500	501	531180	822.2	3321M	51	59k	48k	49k	4529	1	26	1402	
	5.857793e+10		6.053918e+10	3.35%	unknown									
1579s	600	601	663342	905.6	3398M	51	59k	48k	49k	5175	1	36	1426	
	5.857932e+10		6.053918e+10	3.35%	unknown									
L1892s	634	635	704819	922.5	crossove	55	59k	48k	49k	5448	2	41	1433	
	5.858028e+10		6.021605e+10	2.79%	unknown									
i1895s	634	635	715376	939.1	oneopt	55	59k	48k	49k	5448	2	41	1433	
	5.858028e+10		6.021603e+10	2.79%	unknown									
1952s	700	701	770566	929.4	3457M	63	59k	48k	49k	5644	1	50	1442	
	5.858050e+10		6.021603e+10	2.79%	unknown									
2095s	800	801	879949	950.0	3489M	65	59k	48k	49k	5964	1	62	1476	
	5.858065e+10		6.021603e+10	2.79%	unknown									

Файл решения задачи доступен по ссылке <https://disk.yandex.ru/d/KXzdrUx6TZbXEw>

Файл статистической сводки доступен по ссылке <https://disk.yandex.ru/d/FEroaFsr5zbkjA>

**Вывод по сценарию:** описанная выше метаконфигурация приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках для последнего решения из пула допустимых целочисленных решений на 0.97% лучше в смысле целевой функции и на 56.24% – в смысле временных издержек (табл. 15).

В табл. 15 через SCIP+MC (a) обозначается решение, построенное на метаконфигурации SCIP, отвечающее *первому* допустимому целочисленному решению, верхняя граница которого не превышает верхнюю границу решения на доменно-ориентированных эвристиках, а через SCIP+MC (b) – решение, отвечающее *последнему* допустимому целочисленному решению в наборе полученных.

Синим цветом обозначен выигрыш в процентах.

Таблица 15. Сводка результатов анализа эффективности метаконфигурации FZBIVSUHPB. Сценарий f398266b с бинарными переменными

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DOH	72.17	6.080841
SCIP+MC (a)	19.38 +73.15%	6.054807 +0.43%
SCIP+MC (b)	31.58 +56.24%	6.021603 +0.97%

#### 11.2.5. Сценарий 337 с бинарными переменными

##### Статистика

Общее количество переменных: 859230

Количество целочисленных переменных: 173622

Количество бинарных переменных: 155

Количество ограничений: 624637

lp-файл: <https://disk.yandex.ru/d/Kc11p9v7D-kxYA>

##### Анализ решения

Пул решений задачи был найден с помощью следующих первичных эвристик:

- INTSHIFTING,
- RENS,
- ONEOPT.

Фрагмент лога сессии SCIP

time	node	left	LP iter	LP it/n	mem/heur	mdpt	vars	cons	rows	cuts	sepa	confs	strbr
		dualbound	primalbound	gap	compl.								
r 107s	1	0	55407	-	intshift	0	56k	43k	45k	1799	13	0	0
		2.947544e+10	4.344720e+10	47.40%	unknown								
L 247s	1	0	55407	-	rens	0	56k	43k	45k	1799	13	0	0
		2.947544e+10	3.022206e+10	2.53%	unknown								
249s	1	0	55407	-	2785M	0	56k	43k	45k	1799	13	0	0
		2.947544e+10	3.022206e+10	2.53%	unknown								
i 250s	1	0	58839	-	oneopt	0	56k	43k	45k	1799	13	0	0
		2.947544e+10	3.022205e+10	2.53%	unknown								
250s	1	0	58839	-	2809M	0	56k	43k	45k	1799	13	0	0
		2.947544e+10	3.022205e+10	2.53%	unknown								



251s	1	0	58891	-	2813M	0	56k	43k	45k	1820	14	0	0
	2.947544e+10	3.022205e+10	2.53%	unknown									
251s	1	0	58900	-	2813M	0	56k	43k	44k	1824	15	0	0
	2.947544e+10	3.022205e+10	2.53%	unknown									
253s	1	0	59074	-	2816M	0	56k	43k	44k	1824	15	0	12
	2.947544e+10	3.022205e+10	2.53%	unknown									
254s	1	0	59236	-	2821M	0	56k	43k	44k	1918	16	0	12
	2.948327e+10	3.022205e+10	2.51%	unknown									
254s	1	0	59300	-	2821M	0	56k	43k	44k	1945	17	0	12
	2.948327e+10	3.022205e+10	2.51%	unknown									
255s	1	0	59321	-	2821M	0	56k	43k	44k	1945	17	0	19
	2.948327e+10	3.022205e+10	2.51%	unknown									
256s	1	0	59349	-	2825M	0	56k	43k	44k	1959	18	0	19
	2.948327e+10	3.022205e+10	2.51%	unknown									
256s	1	0	59352	-	2825M	0	56k	43k	44k	1964	19	0	19
	2.948327e+10	3.022205e+10	2.51%	unknown									
258s	1	0	59368	-	2825M	0	56k	43k	44k	1964	19	0	35
	2.957927e+10	3.022205e+10	2.17%	unknown									
259s	1	0	59451	-	2829M	0	56k	43k	44k	2014	20	0	35
	2.957927e+10	3.022205e+10	2.17%	unknown									
259s	1	0	59466	-	2829M	0	56k	43k	44k	2024	21	0	35
	2.957927e+10	3.022205e+10	2.17%	unknown									
259s	1	2	59466	-	2829M	0	56k	43k	44k	2024	21	0	35
	2.957927e+10	3.022205e+10	2.17%	unknown									

Файл решения задачи доступен по ссылке <https://disk.yandex.ru/d/zwVhKYKEMlMlQw>

Файл статистической сводки доступен по ссылке <https://disk.yandex.ru/d/T9sAbRH6uWh4Uw>

**Вывод по сценарию:** описанная выше метаконфигурация приводит к решению задачи, которое оказывается по отношению к результату на доменно-ориентированных эвристиках для последнего решения из пула допустимых целочисленных решений на ...% лучше в смысле целевой функции и на ...% – в смысле временных издержек (табл. 16).

В табл. 16 через SCIP+MC (a) обозначается решение, построенное на метаконфигурации SCIP, отвечающее *первому* допустимому целочисленному решению, верхняя граница которого не превышает верхнюю границу решения на доменно-ориентированных эвристиках, а через SCIP+MC (b) – решение, отвечающее *последнему* допустимому целочисленному решению в наборе полученных.

Синим цветом обозначен выигрыш в процентах.

Таблица 16. Сводка результатов анализа эффективности метаконфигурации FZBIVSUNPB. Сценарий 337 с бинарными переменными

Способ	Полное время рас- чета, мин	Верхняя граница решения, $\times 10^{10}$
CBC+DON	18.00	4.047865
SCIP+MC (a)	4.12 +77.11%	3.022206 +25.34%
SCIP+MC (b)	4.30 +76.11%	3.022205 +25.34%

### 11.3. Поиск решения на базе методов машинного и глубокого обучения

Условимся *сценарием обучающего поднабора* называть сценарий (математическую постановку задачи, описанную в терминах математического программирования) из коллекции сценариев, которые используются на *обучающей фазе* модели машинного обучения.



Сценарий input\_a78cbead-073b-ec11-a2d4-005056a5ee74.json (1-ая и 2-ая фазы поиска решения)

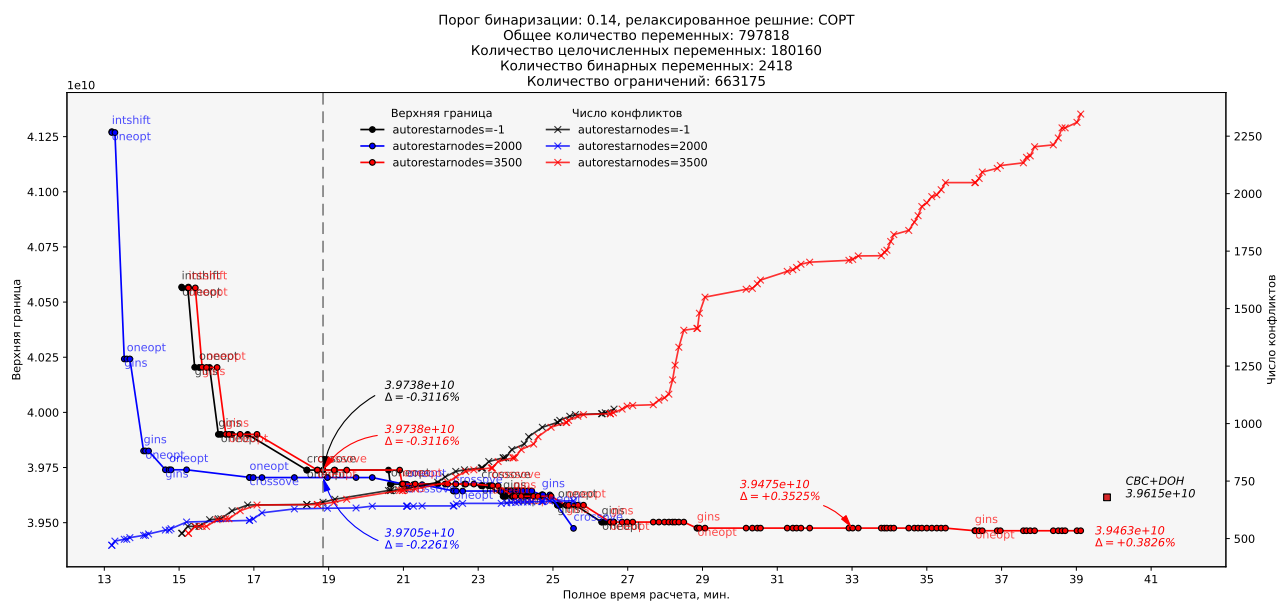


Рис. 22. Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра autorestartnodes. Сценарий input\_a78cbead. Первая и вторая фазы поиска решения

Сценарий input\_a78cbead-073b-ec11-a2d4-005056a5ee74.json (3-ья фаза поиска решения)

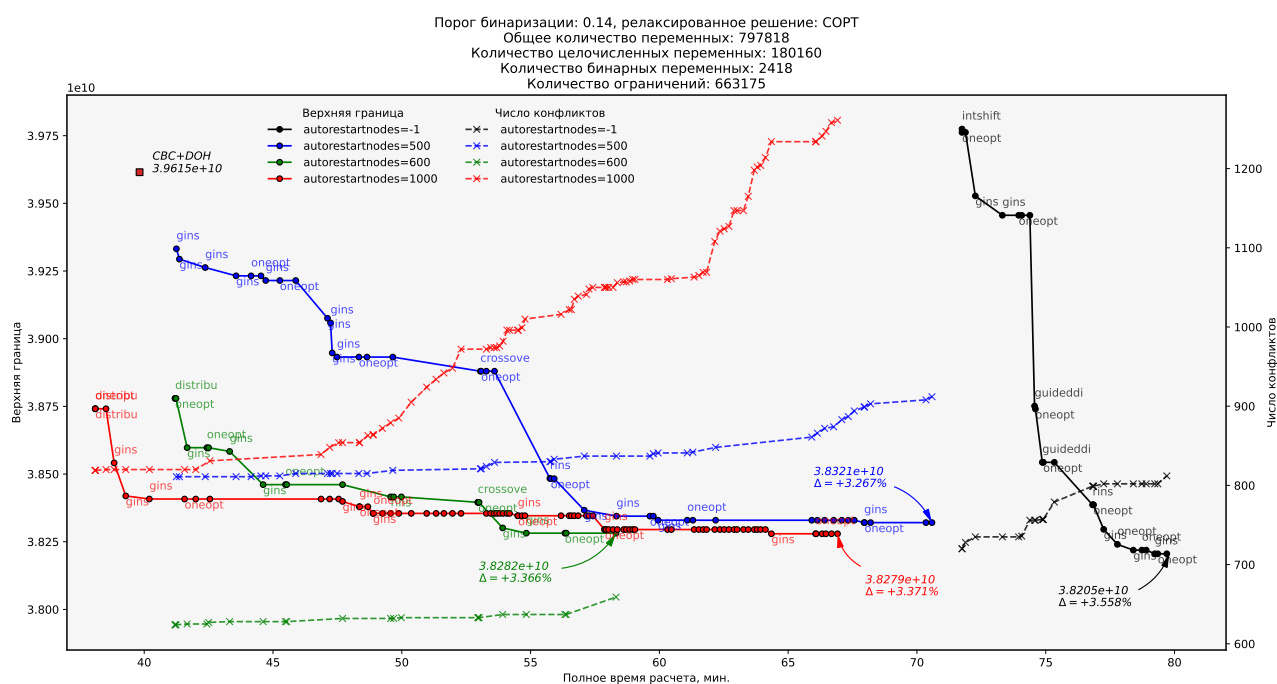


Рис. 23. Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра autorestartnodes. Сценарий a78cbead. Третья фаза поиска решения

Сценарием тестового поднабора условимся называть сценарий, который используется для построения прогноза с помощью модели машинного обучения.

### 11.3.1. Простое декартово произведение сценариев с бинарными переменными

Рассмотрим некоммутативные пары вида «сценарий обучающего поднабора – сценарий тестового поднабора» подгруппы сценариев с бинарными переменными (см. раздел 11.2):



обучение на сценарии 7fac4231\_bin.lp, тестирование на сценарии 50197df7\_bin.lp ...  
обучение на сценарии a78cbead\_bin.lp, тестирование на сценарии 50197df7\_bin.lp ...  
обучение на сценарии f398266b\_bin.lp, тестирование на сценарии 50197df7\_bin.lp ...  
обучение на сценарии 337\_bin.lp, тестирование на сценарии 50197df7\_bin.lp ...провал  
обучение на сценарии 7fac4231\_bin.lp, тестирование на сценарии 50197df7\_bin.lp

## **12. Описание вычислительных экспериментов на сценариях группы MBO**

## **13. Описание вычислительных экспериментов на сценариях MIPLIB 2017**

### **13.1. Сценарии со статусом «open»**

#### **13.1.1. Сценарий DLR2**

<https://miplib.zib.de/WebData/instances/dlr2.mps.gz>

#### **13.1.2. Сценарий CVRPA-N64K9VRPI**

<https://miplib.zib.de/WebData/instances/cvrpa-n64k9vrpi.mps.gz>

### **13.2. Сценарии со статусом «hard»**

#### **13.2.1. Сценарий CRYPTANALYSISKB128N5OBJ14**

<https://miplib.zib.de/WebData/instances/cryptanalysiskb128n5obj14.mps.gz>

### **13.3. Сценарии со статусом «easy»**

#### **13.3.1. Сценарий NEOS-4332801-seret**

<https://miplib.zib.de/WebData/instances/neos-4332801-seret.mps.gz>

## Список иллюстраций

1	Сводка результатов вычислительных экспериментов на сценариях группы ИКП . . . . .	16
2	Важность признаков по Шепли . . . . .	19
3	График частичной зависимости признака <code>var_ub_global_after_presolving</code> от признака <code>integer_context_after_presolving</code> . . . . .	19
4	График частичной зависимости признака <code>integer_context_after_presolving</code> от признака <code>var_context_mean_relax_after_presolving</code> . . . . .	20
5	График частичной зависимости признака <code>relax_sol</code> от признака <code>var_ub_global_after_presolving</code> . . . . .	20
6	Локальная интерпретация для первого экземпляра валидационного поднабора . . . . .	21
7	Зависимость верхней границы решения от доли бинарных переменных, выставленных в ноль. Сценарий <code>a78cbead</code> . . . . .	23
8	Зависимость верхней границы решения от доли бинарных переменных, выставленных в ноль. Сценарий <code>337</code> . . . . .	23
9	Сводка результатов вычислительных экспериментов на сценарии группы СОП <code>tmpfvpqodxw.lp</code> . . . . .	29
10	Сводка результатов вычислительных экспериментов на сценарии группы СОП <code>1664182546_82382.lp</code> . . . . .	30
11	Сводка результатов вычислительных экспериментов на сценарии группы СОП <code>1664182533_1587787.lp</code> . . . . .	31
12	Сводка результатов вычислительных экспериментов на сценарии группы СОП <code>1664182480_4326847.lp</code> . . . . .	33
13	Сводка результатов вычислительных экспериментов на сценарии группы СОП <code>1664182523_380519.lp</code> . . . . .	34
14	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>f398266b</code> без бинарных переменных . . . . .	36
15	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>50197df7</code> без бинарных переменных . . . . .	37
16	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>7fac4231</code> без бинарных переменных . . . . .	39
17	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>ca485a55</code> без бинарных переменных . . . . .	40
18	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>276</code> без бинарных переменных . . . . .	41
19	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>337</code> без бинарных переменных . . . . .	43
20	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>13d686ab</code> без бинарных переменных . . . . .	44
21	Сводка результатов анализа эффективности метаконфигурации SUN. Сценарий <code>a78cbead</code> без бинарных переменных . . . . .	45
22	Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра <code>autorestartnodes</code> . Сценарий <code>input_a78cbead</code> . Первая и вторая фазы поиска решения . . . . .	57

23	Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра <code>autorestartnodes</code> . Сценарий <code>a78cbead</code> . Третья фаза поиска решения . . . . .	57
24	Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра <code>autorestartnodes</code> . Сценарий <code>50197df7</code> . Третья фаза поиска решения . . . . .	58
25	Динамика изменения верхней границы решения и числа конфликтов во времени в зависимости от значения параметра <code>autorestartnodes</code> . Сценарий <code>7fac4231</code> . Третья фаза поиска решения . . . . .	58

## Список таблиц

1	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>f398266b</code> без бинарных переменных . . . . .	36
2	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>50197df7</code> без бинарных переменных . . . . .	37
3	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>7fac4231</code> без бинарных переменных . . . . .	38
4	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>ca485a55</code> без бинарных переменных . . . . .	40
5	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>276</code> без бинарных переменных . . . . .	41
6	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>337</code> без бинарных переменных . . . . .	42
7	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>13d686ab</code> без бинарных переменных . . . . .	43
8	Сводка результатов анализа эффективности метаконфигураций <code>SUN</code> и <code>FZBIVSUNPB</code> . Сценарий <code>a78cbead</code> без бинарных переменных . . . . .	45
9	Сводка результатов анализа эффективности метаконфигураций <code>FZBIVSUNPB</code> и ансамбля детекторов аномалий. Сценарий <code>496</code> без бинарных переменных . . . . .	46
10	Сводка результатов анализа эффективности метаконфигураций <code>FZBIVSUNPB</code> и ансамбля детекторов аномалий. Сценарий <code>514</code> без бинарных переменных . . . . .	47
11	Сводка результатов анализа эффективности метаконфигураций <code>FZBIVSUNPB</code> и ансамбля детекторов аномалий. Сценарий <code>519</code> без бинарных переменных . . . . .	48
12	Сводка результатов анализа эффективности метаконфигурации <code>FZBIVSUNPB</code> . Сценарий <code>a78cbead</code> с бинарными переменными . . . . .	51
13	Сводка результатов анализа эффективности метаконфигурации <code>FZBIVSUNPB</code> . Сценарий <code>7fac4231</code> с бинарными переменными . . . . .	52
14	Сводка результатов анализа эффективности метаконфигурации <code>FZBIVSUNPB</code> . Сценарий <code>50197df7</code> с бинарными переменными . . . . .	54
15	Сводка результатов анализа эффективности метаконфигурации <code>FZBIVSUNPB</code> . Сценарий <code>f398266b</code> с бинарными переменными . . . . .	55
16	Сводка результатов анализа эффективности метаконфигурации <code>FZBIVSUNPB</code> . Сценарий <code>337</code> с бинарными переменными . . . . .	56

## Список литературы

1. *Иванов* Конспект по обучению с подкреплением, 2022
2. *Жерон, О.* Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow: концепции, инструменты и техники для создания интеллектуальных систем. – СПб.: ООО «Альфа-книга», 2018. – 688 с.
3. *Soenen J. etc.* The Effect of Hyperparameter Tuning on the Comparative Evaluation of Unsupervised Anomaly Detection Methods, 2021