**АННОТАЦИЯ**

Выпускная квалификационная работа магистра “Планирование виртуальных машин в распределенных системах при выполнении заданий”

Пояснительная записка: 117 страниц, 26 источников, 1 приложение.

Объектом исследования является процессы распределённого хранения и обработки данных.

Предметом исследования является математическая модель распределенного хранения обработки и методы оптимизации решений по распределённому хранению и распределённой обработки данных.

Целью работы является совершенствование математических моделей частично целочисленного программирования с целью решения задачи по управлению распределенного хранения и распределенной обработки данных

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc138143171)

[1.АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ, НАПРАВЛЕННЫХ НА ОПТИМИЗАЦИЮ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ И РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ 9](#_Toc138143172)

[1.1 Анализ моделей, направленных на оптимизацию распределенного хранения и распределенной обработки данных при минимизации стоимости 9](#_Toc138143173)

[1.2 Анализ моделей, направленных на оптимизацию распределенного хранения и распределенной обработки данных при минимизации длительности обработки 20](#_Toc138143174)

[2. ОБОСНОВАНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СМЕШАННОГО ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ОПТИМИЗАЦИИ РЕШЕНИЙ ПО РАСПРЕДЕЛЕННЫМ ХРАНЕНИЯМ ДАННЫХ И РАСПРЕДЕЛЕННОЙ ИХ ОБРАБОТКИ 28](#_Toc138143175)

[2.1 Обоснование математической модели смешанного целочисленного линейного программирования оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения операций и ограничениях на директивные сроки обработки 28](#_Toc138143176)

[2.2 Обоснование математической модели смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения 36](#_Toc138143177)

[3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РЕШЕНИЯ ЗАДАЧ СМЕШАННОГО ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ 43](#_Toc138143178)

[3.1. Выбор программного обеспечения 43](#_Toc138143179)

[3.2 Описание функций, используемых в приложении Or-toolls, и их атрибутов используемых при решении задач 48](#_Toc138143180)

[3.3 Описание синтаксических конструкций 51](#_Toc138143181)

[3.4 Описание программы 52](#_Toc138143182)

[4. АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ МОДЕЛЕЙ, НАПРАВЛЕННЫХ НА ОПТИМИЗАЦИЮ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ И РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ 58](#_Toc138143183)

[4.1. Анализ первой математической модели смешанного целочисленного линейного программирования оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения операций и ограничениях на директивные сроки обработки. 58](#_Toc138143184)

[4.2. Анализ второй математической модели смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения 84](#_Toc138143185)

[ЗАКЛЮЧЕНИЕ 112](#_Toc138143186)

[СПИСОК ЛИТЕРАТУРЫ 113](#_Toc138143187)

[ПРИЛОЖЕНИЕ А. КОД ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 116](#_Toc138143188)

# **ВВЕДЕНИЕ**

**Актуальность.** С развитием информационных технологий и повышением необходимости в обработке больших массивов разнотипных данных, в современном мире всё большую популярность набирают распределённые системы обработки информации. Такие системы получили широкое распространение и используются для решения целого спектра задач. Процесс организации вычислений в таких системах предполагает решения ряда проблем, включающих в себя распределение данных поступаемых на вход системы заданий, а также назначения приложений, обрабатывающих эти данные, по вычислительным устройствам [1].

Оптимизация решений по хранению и обработке данных позволяет распределить данные по нескольким узлам и использовать ресурсы каждого узла наиболее эффективно. Позволяет снизить время отклика на вопросы пользователей связанное с обработкой данных.

Каналы передачи данных играют важную роль в распределенных системах, так как они определяют скорость передачи данных между узлами. Необходимо учитывать стоимость каналов передачи данных, чтобы оптимизировать затраты на создание и поддержку распределенной системы хранения и обработки данных.

Стоимость также является важным фактором в распределенных системах, поскольку использование большого количества вычислительных ресурсов может стоить дорого. Планирование виртуальных машин может помочь управлять затратами на вычислительные ресурсы, распределяя нагрузку между серверами более эффективно.

Виртуальная машина - это абстрактная модель вычислительной системы, которая позволяет запускать программы на нескольких физических машинах. Виртуальная машина может использоваться для обработки запросов пользователей в распределенной системе хранения и обработки данных, так как она позволяет эффективно использовать ресурсы каждого узла системы.

Вследствие того, что размерность входных параметров для таких систем может быть достаточно большой (количество типов, обрабатываемых данных, вычислительный устройств), большинство подходов, направленных на получение оптимального решения, не могут быть применены в рамках предметной области, по причине того, что такое решение не может быть получено за разумное время. По этой причине широкое распространение на практике получили приближенные алгоритмы, не гарантирующие нахождение оптимального решения, однако способные получить достаточно эффективное решение за значительно меньшее время. Несмотря на свою эффективность, такие методы требуют гибкой настройки под конкретно решаемую задачу. Совершенствование таких методов для решения задач планирования распределений ресурсов и выполнения заданий в распределённых системах обработки данных является актуальной.

**Объектом** является процессы распределённого хранения, передачи и обработки данных.

**Предметом** является математическая модель распределенного хранения обработки и методы оптимизации решений по распределённому хранению и распределённой обработке данных.

**Целью** является совершенствование математических моделей частично целочисленного линейного программирования с целью решения задачи по управлению распределенного хранения и распределенной обработки данных

Для достижения поставленной цели следует решить следующие задачи:

1. Обоснование актуальности темы исследования;
2. Анализ существующих моделей смешанного целочисленного программирования, недостатков, возможностей их совершенствования;
3. Формирования целочисленной модели.

**Применение.** Разработанные способы оптимизации могут быть применены в области облачных вычислений. Рассмотрим следующие механизмы:

1. Распределенное хранение данных: Облачные вычисления используют распределенное хранение данных для хранения данных клиентов в облаке. Данные хранятся на различных серверах, которые могут находиться в разных местах мира. Распределенное хранение данных обеспечивает отказоустойчивость системы, т.к. если один сервер выходит из строя, данные остаются доступными на других серверах. Это также обеспечивает высокую доступность данных, т.к. данные можно получить быстрее, так как они распределены между различными серверами;

2. Распределенная обработка данных: В облачных вычислениях распределенная обработка данных используется для обработки данных клиентов. Например, компании могут использовать облачные вычисления для обработки больших объемов данных, которые могут быть слишком большими для обработки на локальном компьютере или сервере. Распределенная обработка данных обычно выполняется на кластере серверов, где каждый сервер обрабатывает свою часть данных. Это позволяет ускорить обработку данных и выполнить больший объем работы в короткие сроки;

3. Хранение и обработка данных в реальном времени: Распределенное хранение и обработка данных также используется в области обработки данных в реальном времени. Компании могут использовать облачные вычисления для анализа данных в реальном времени, чтобы получить быстрый ответ на изменения в условиях рынка или изменения в поведении клиентов. Для этого используется распределенная обработка данных в режиме реального времени, где данные обрабатываются немедленно, как только они поступают;

4. Снижение затрат: Использование распределенного хранения и обработки данных в облачных вычислениях позволяет компаниям снизить затраты на оборудование, т. к. как правило, облачные провайдеры предоставляют гибкую модель оплаты, где клиенты могут оплачивать только те ресурсы, которые они фактически используют. Это позволяет компаниям снизить свои операционные затраты, так как им не нужно приобретать и поддерживать собственное оборудование;

5. Масштабируемость: Распределенное хранение и обработка данных в облачных вычислениях также обеспечивает масштабируемость, что позволяет компаниям масштабировать свою инфраструктуру в соответствии с растущими потребностями. Компании могут легко добавлять новые серверы и хранилища данных в облаке, чтобы обеспечить масштабируемость своих приложений и услуг;

6. Географическое распределение данных: Распределенное хранение данных также позволяет компаниям хранить данные клиентов в различных географических регионах, что повышает скорость доступа к данным и снижает задержки в обработке. Это особенно важно для компаний, которые имеют клиентов в различных частях мира.

# **1.АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ, НАПРАВЛЕННЫХ НА ОПТИМИЗАЦИЮ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ И РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ**

## **1.1 Анализ моделей, направленных на оптимизацию распределенного хранения и распределенной обработки данных при минимизации стоимости**

В настоящее время активно развиваются математические модели процессов распределённого хранения и обработке данных, методы оптимизации решений по распределенному хранению и обработки данных.

В работе [2] рассмотрена модель процесса распределенного хранения и обработки данных, использование которой позволяет минимизировать стоимость размещения каждой пары виртуальных машин v и w в центре обработке данных (ЦОД) i и j.

Для построения модели введены следующие обозначения:

N – множество центров обработки данных; K – множество виртуальных машин; v – множество пользователей; – вместимость виртуальных машин в ЦОД i; – пропускная способность между ЦОД i и j; – задержка между ЦОД i и j; – стоимость передачи единицы данных между ЦОД i и j; – требуемая пропускная способность между виртуальными машинами v и w; – требуемая задержка между виртуальными машинами v и w; d(u) – центр обработки данных, в котором находится пользователь u; – требуемая задержка между пользователем u и виртуальной машиной v; – стоимость размещения виртуальной машины v в ЦОД i; – двоичная переменная устанавливается равной единице, когда VM v находится в ЦОД i, и нулю в противном случае; – это переменная, которая отвечает за количество переданных данных между виртуальными машинами v и w, которые находятся в разных центрах обработки данных j и w соответственно. z – коэффициент стоимости масштабирования.

Целевая функция в модели [2] имеет следующим вид:

В модели [2] были введены следующие ограничения:

(1.1)

(1.2)

(1.3)

(1.4)

(1.5)

(1.6)

Ограничения (1.1) требуют, чтобы количество виртуальных машин в каждом ЦОД не превышало его вместимости.

Ограничения (1.2) требуют, чтобы каждая виртуальная машина была назначена ровно в одном ЦОД.

Ограничения (1.3) требуют, чтобы заданная пропускная способность между каждой парой ЦОД i и j не была превышена общей суммой пропускной способности, требуемой для всех виртуальных машин, размещенных в этих ЦОД.

Ограничения (1.4) гарантируют, что требуемая задержка между каждой парой виртуальных машин должна соблюдаться, то есть, если виртуальные машины v и w размещены соответственно в ЦОД i и j, то задержка между ЦОД i и j не должна превышать требуемой задержки между виртуальными машинами v и w.

Ограничения (1.5) требуют, чтобы задержка между виртуальной машиной v и ЦОД, где находится пользователь u, соблюдалась, то есть, виртуальная машина v может быть размещена только на ЦОД i, если задержка между i и d(u) меньше или равна заданной задержке между виртуальной машиной v и пользователем u.

Ограничения (1.6) определяют значения переменных.

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов, объемы хранимых и обрабатываемых этими приложениями данных, производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных.

В работе [2] рассмотрена модель процесса распределенного хранения и обработки данных, использование которой позволяет минимизировать стоимость размещения каждой пары виртуальных машин v и w в центре обработке данных (ЦОД) i и j.

Для построения модели введены следующие обозначения:

Для построения модели введены следующие обозначения: N – множество центров обработки данных; K – множество виртуальных машин; v – множество пользователей; – вместимость виртуальных машин в ЦОД i;

– пропускная способность между ЦОД i и j; – задержка между ЦОД i и j; – стоимость передачи единицы данных между ЦОД i и j; – требуемая пропускная способность между виртуальными машинами v и w; – требуемая задержка между виртуальными машинами v и w; d(u) – центр обработки данных, в котором находится пользователь u; – требуемая задержка между пользователем u и виртуальной машиной v; – стоимость размещения виртуальной машины v в ЦОД i; – двоичная переменная устанавливается равной единице, когда VM v находится в ЦОД i, и нулю в противном случае; – это переменная, которая отвечает за количество переданных данных между виртуальными машинами v и w, которые находятся в разных центрах обработки данных j и w соответственно; yivjw и yjwiv - переменные, которые соответствуют пропускной способности между парами виртуальных машин v и w, размещенных в разных центрах данных i и j

z – коэффициент стоимости масштабирования.

Для линеаризации модели 1.1 – 1.6 в рассмотрении введена переменная y значения которых вычисляется следующим образом: yivjw = xivxjw, ∀ i, j = {1, . . ., N} и v, w = {1, . . ., K}.

Линейная математическая модель со смешанным целым числом для VMPlacement, названная LMVMP [3], может быть сформулирована следующим образом:

LMVMP получается путем замены произведения xivxjw на yivjw из QMVMP (квадратичная модель размещения виртуальной машины).

В модели [2] были введены следующие ограничения:

(1.11)

(1.14)

(1.13)

(1.12)

(1.8)

(1.9)

(1.10)

(1.7)

(1.15)

Ограничения (1.7) требуют, чтобы количество виртуальных машин в каждом ЦОД не превышало его вместимости.

Ограничения (1.8) требуют, чтобы каждая виртуальная машина была назначена ровно в одном ЦОД.

Ограничения (1.9) и (1.10) определяют связь между переменными x и y.

Ограничения (1.11) также накладывают симметричное соотношение на переменные y.

Ограничения (1.12) требуют, чтобы заданная пропускная способность между каждой парой ЦОД i и j не была превышена общей суммой пропускной способности, требуемой для всех виртуальных машин, размещенных в этих ЦОД.

Ограничения (1.13) гарантируют, что требуемая задержка между каждой парой виртуальных машин должна соблюдаться, то есть, если виртуальные машины v и w размещены соответственно в ЦОД i и j, то задержка между ЦОД i и j не должна превышать требуемой задержки между виртуальными машинами v и w.

Ограничения (1.14) требуют, чтобы задержка между виртуальной машиной v и ЦОД, где находится пользователь u, соблюдалась, то есть, виртуальная машина v может быть размещена только на ЦОД i, если задержка между i и d(u) меньше или равна заданной задержке между виртуальной машиной v и пользователем u.

Ограничения (1.15) определяют диапазон переменных.

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов, объемы хранимых и обрабатываемых этими приложениями данных, производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных.

В работе [4] была рассмотрена модель процесса распределенного хранения и обработки данных, использование которой позволяет минимизировать общие затраты, связанные с размещением виртуальных машин в распределенных центрах обработки данных.

Для построения модели введены следующие обозначения: N –множество центров обработки данных; K – множество виртуальных машин;

v – множество пользователей; – вместимость виртуальных машин в ЦОД i;

– пропускная способность между ЦОД i и j; – задержка между ЦОД i и j; – стоимость передачи единицы данных между ЦОД i и j; – требуемая пропускная способность между виртуальными машинами v и w; – требуемая задержка между виртуальными машинами v и w; d(u) – центр обработки данных, в котором находится пользователь u; – требуемая задержка между пользователем u и виртуальной машиной v; – стоимость размещения виртуальной машины v в ЦОД i; – двоичная переменная устанавливается равной единице, когда VM v находится в ЦОД i, и нулю в противном случае; yivjw и yjwiv – переменные, которые соответствуют пропускной способности между парами виртуальных машин v и w, размещенных в разных центрах данных i и j

– бинарная переменная, которая равна 1, если передача данных от пользователей к центрам обработки данных v, а затем к виртуальной машине i, проходит через центры обработки данных v и w; z – коэффициент затрат на масштабирование.

В целевой функции, предложенной в [7], первое слагаемое отвечает за затраты на размещение виртуальных машин в центрах обработки данных, а второе слагаемое учитывает затраты на связь между центрами обработки данных. Переменная z является коэффициентом затрат на масштабирование.

Ограничения включают в себя ограничения на емкость центра обработки данных, пропускную способность и задержку между центрами обработки данных, а также ограничение на задержку между пользователем и виртуальной машиной.

(1.21)

(1.20)

(1.19)

(1.18)

(1.17)

(1f)

(1d)

(1e)

(1с)

(1.21)

(1.20)

(1.19)

(1.18)

(1.17)

(1.16)

Ограничения (1.16) требуют, чтобы количество виртуальных машин в каждом ЦОД не превышало его вместимости.

Ограничения (1.17) требуют, чтобы каждая виртуальная машина была назначена ровно в одном ЦОД.

Ограничения (1.18) на пропускную способность между центрами обработки данных

Ограничения (1.19) гарантирует, что задержка между виртуальными машинами i и j не будет превышать допустимой задержки.

Ограничение (1.20) гарантирует, что задержка между пользователем и его виртуальной машиной не будет превышать допустимую задержку m.

Ограничение (1.21): и , ограничены бинарными значениями {0, 1}, что означает, что виртуальные машины могут быть размещены только в одном месте, передача данных может происходить только между соответствующими парой виртуальных машин, и передача данных между пользователями и виртуальными машинами может проходить через определенные центры обработки данных.

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов, объемы хранимых и обрабатываемых этими приложениями данных, производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных.

В работе [5] была рассмотрена модель процесса распределенного хранения и обработки данных, использование которой позволяет минимизировать общую стоимость решения, суммарное время выбранных решений, с учетом их производительности.

Для построения модели введены следующие обозначения: N – количество виртуальных машин; M – количество центров обработки данных;

– 1, если виртуальная машина i размещена на центре обработки данных j, иначе 0 (двоичная переменная); – затраты на связь между виртуальной машиной i и центром обработки данных j; – выделенная единица объема памяти для виртуальной машины i; – объем данных, обрабатываемых виртуальной машиной i; – производительность вычислений для виртуальной машины I; – доступная вычислительная мощность на центре обработки данных j; – доступное отношение объема выделенной памяти к производительности вычислений на центре обработки данных j; – минимальный процент загрузки центра обработки данных j.

Целевая функция представляет задачу целочисленного линейного программирования и состоит из трех частей, которые связаны между собой суммированием.

Cуммарная стоимость всех выбранных решений (), где каждое решение () имеет свою стоимость (). Эта часть целевой функции отвечает за минимизацию общей стоимости решения:

Cуммарный объем выбранных решений, где каждое решение имеет свой объем. Эта часть целевой функции отвечает за максимизацию суммарного объема выбранных решений.

Cуммарное время выбранных решений, где каждое решение имеет свое время и делится на его производительность. Эта часть целевой функции отвечает за минимизацию суммарного времени выбранных решений, с учетом их производительности:

Итоговая целевая функция выглядит следующим образом:

В модели [5] были введены следующие ограничения:

Для всех i (каждая виртуальная машина должна быть размещена только на одном центре обработки данных):

Для всех j (выделенная единица объема памяти для виртуальных машин на каждом центре обработки данных должна быть не больше общего объема памяти на центре обработки данных):

Для всех j (объем данных, обрабатываемых виртуальными машинами на каждом центре обработки данных должен быть не больше доступного объема данных на центре обработки данных):

Для всех j (производительность вычислений для виртуальных машин на каждом центре обработки данных должна быть не больше доступной производительности на центре обработки данных):

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов.

В работе [6] была рассмотрена модель процесса распределенного хранения и обработки данных, использование которой позволяет минимизировать суммарную стоимость и производительность выбранных решений

Для построения модели введены следующие обозначения:

N – набор виртуальных машин; M – набор центров обработки данных; pi – производительность вычислений для виртуальной машины i; mi – объем памяти; qi – объем запросов клиентов; qj – это ограничение на общий объем запросов клиентов, для всех размещенных на центре обработки данных j виртуальных машин; dk,j –расстояние между ЦОД k и j; xk,j – это переменная, отвечающая за количество ресурсов, выделенных для обработки задачи j на сервере k; xi,j – это булева переменная, которая принимает значение 1, если виртуальная машина i размещена на центре обработки данных j, и 0 в противном случае; vj – это ограничение на общий объем памяти, который может быть выделен на центре обработки данных j для всех размещенных на нем виртуальных машин; Cj – максимальное количество ВМ, которое может быть размещено на ЦОД j; cj – это стоимость использования связи между серверами j и k.

Целевая функция представляет задачу целочисленного линейного программирования и состоит из двух частей, которые связаны между собой суммированием.

Cуммарная производительность всех выбранных решений, где каждое решение имеет свою производительность для заданного i. Эта часть целевой функции отвечает за максимизацию суммарной производительности выбранных решений:

Суммарная стоимость всех выбранных решений, где каждое решение имеет свою стоимость и время выполнения. Эта часть целевой функции отвечает за минимизацию суммарной стоимости выбранных решений:

Итоговая целевая функция выглядит следующим образом:

В модели [6] были введены следующие ограничения:

Каждая ВМ должна быть размещена на одном из ЦОД:

Общий объем памяти, выделенной для всех ВМ на ЦОД j, не должен превышать выделенную единицу объема памяти на ЦОД:

Общий объем запросов клиентов для всех ВМ на ЦОД j не должен превышать заданный порог:

Ограничение на количество ВМ, размещенных на каждом ЦОД:

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов.

## **1.2 Анализ моделей, направленных на оптимизацию распределенного хранения и распределенной обработки данных при минимизации длительности обработки**

В работе [7] была рассмотрена модель процесса распределенного хранения и обработки данных, позволяющая минимизировать суммарное время выполнения выбранных решений, суммарную задержку выполнения выбранных решений, штраф за невыбранные решения, количество невыбранных решений, дополнительный штраф за невыбранные решения с высокой производительностью и количество невыбранных решений с высокой производительностью.

Для построения модели были введены следующие обозначения: cij – затраты на связь между центрами обработки данных i и j; mi – затраты на выделенное место для виртуальной машины i на центре обработки данных; di – объем данных, которые нужно обрабатывать виртуальной машине I; pi – производительность вычислений для виртуальной машины i; xij - переменная решения, принимающая значение 1, если виртуальная машина i размещена на центре обработки данных j, и 0 в противном случае; α – коэффициент, учитывающий общее число размещенных виртуальных машин; β – коэффициент, учитывающий объем данных, обрабатываемых на каждом центре обработки данных; qi – запрос клиента, связанный с объемом данных, который обрабатывается виртуальной машиной I; Qj – доступная емкость на центре обработки данных j; Mj – доступная вычислительная мощность на центре обработки данных j; Dj – доступное отношение объема выделенной памяти к производительности вычислений на центре обработки данных j; Lj –минимальный процент загрузки центра обработки данных j.

Целевая функция представляет задачу целочисленного линейного программирования и состоит из нескольких частей, которые связаны между собой суммированием.

Суммарная стоимость выбранных решений , где каждое решение имеет свою стоимость . Эта часть целевой функции отвечает за минимизацию суммарной стоимости выбранных решений:

Суммарное время выполнения всех выбранных решений , где каждое решение имеет свое время выполнения . Эта часть целевой функции отвечает за минимизацию суммарного времени выполнения выбранных решений:

Суммарная задержка выполнения всех выбранных решений , где каждое решение имеет свою задержку выполнения и производительность . Эта часть целевой функции отвечает за минимизацию суммарной задержки выполнения выбранных решений:

Штраф за невыбранные решения, где α - коэффициент штрафа:

Количество невыбранных решений. Эта часть целевой функции отвечает за стимулирование выбора как можно большего числа решений:

Дополнительный штраф за невыбранные решения с высокой производительностью, где β - коэффициент дополнительного штрафа

Количество невыбранных решений с высокой производительностью ( - производительность решения i). Эта часть целевой функции отвечает за стимулирование выбора решений с высокой производительностью:

Итоговая целевая функция выглядит следующим образом:

В модели [7] были введены следующие ограничения:

Для всех i,j - каждая виртуальная машина может быть размещена только на одном центре обработки данных:

Для всех j - общий объем данных, обрабатываемых на каждом центре обработки данных, не должен превышать доступную ему емкость:

Для всех j - общая вычислительная мощность, занимаемая каждой виртуальной машиной, не должна превышать доступную вычислительную мощность на каждом центре обработки данных:

Для всех j - суммарное отношение объема выделенной памяти к производительности вычислений для всех виртуальных машин, размещенных на каждом центре обработки данных, не должно превышать доступное значение на каждом центре обработки данных:

Для всех j - каждый центр обработки данных должен быть загружен не менее, чем на определенный процент (Lj) его общей емкости:

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов.

В работе [8] была рассмотрена модель процесса распределенного хранения и обработки данных, позволяющая минимизировать затраты на размещение виртуальных машин в центрах обработки данных, а также затраты на выделение ресурсов для каждой виртуальной машины.

Для построения модели были введены следующие обозначения:

n – количество виртуальных машин; m – количество центров обработки данных; сi,j – стоимость размещения i-й виртуальной машины в j-м центре обработки данных; xij – бинарная переменная, равная 1, если i-я виртуальная машина размещена в j-м центре обработки данных, иначе 0; bi – затраты на выделенную единицу объема памяти и запросов клиентов по объему данных и производительности вычислений для i-й виртуальной машины; yi – непрерывная переменная, равная количеству ресурсов, выделенных для i-й виртуальной машины; C – общий бюджет на выделение ресурсов; si –требуемое количество памяти и ресурсов для i-й виртуальной машины; Si – максимальное количество памяти и ресурсов, которые могут быть выделены для i-й виртуальной машины; di – объем данных, обрабатываемых i-й виртуальной машиной

Целевая функция состоит из двух частей: первая часть выражает затраты на размещение виртуальных машин в центрах обработки данных, а вторая часть выражает затраты на выделение ресурсов для каждой виртуальной машины.

В модели [8] были введены следующие ограничения:

(1.25)

(1.24)

(1.23)

(1.22)

Первое ограничение гарантирует, что каждая виртуальная машина будет размещена только в одном центре обработки данных.

Второе ограничение гарантирует, что общее количество ресурсов, выделенных для всех виртуальных машин, не превышает общий бюджет.

Третье ограничение гарантирует, что для каждой виртуальной машины выделяется не менее требуемого количества ресурсов.

Четвертое ограничение гарантирует, что объем данных, обрабатываемых всеми виртуальными машинами, не превышает максимально допустимый уровень, обозначенный как D.

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов, производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных.

В работе [9] была рассмотрена модель процесса распределенного хранения и обработки данных, позволяющая минимизировать затраты на связь между виртуальной машиной, затраты на выделении памяти, затраты на обработку запросов клиентов.

Для построения модели были введены следующие обозначения: bi – максимальный объем памяти для виртуальной машины i; xim – бинарная переменная, равная 1, если виртуальная машина i размещена на центре обработки данных m и 0 в противном случае; yim – непрерывная переменная, обозначающая объем памяти, выделенный для виртуальной машины i на центре обработки данных m; zim – непрерывная переменная, обозначающая количество запросов, поступающих от клиентов, для виртуальной машины i на центре обработки данных m; cim – затраты на связь между виртуальной машиной i и центром обработки данных m; lim – затраты на выделенную единицу объема памяти для виртуальной машины i на центре обработки данных m; qim – затраты на обработку запросов клиентов для виртуальной машины i на центре обработки данных m.

Целевая функция описывает задачу размещения виртуальных машин на центрах обработки данных. Цель состоит в минимизации затрат на размещение виртуальных машин на центрах обработки данных. Она состоит из трех частей: первая часть отвечает за затраты на связь между виртуальной машиной и центром обработки данных, вторая часть - за затраты на выделение памяти, третья часть - за затраты на обработку запросов клиентов.

В модели [9] были введены следующие ограничения:

Каждая виртуальная машина должна быть размещена только на одном центре обработки данных:

Общий объем памяти, выделенный для каждой виртуальной машины на всех центрах обработки данных, не должен превышать максимальный объем памяти для каждой виртуальной машины:

Количество запросов, поступающих от клиентов, должно быть равным количеству запросов, обрабатываемых виртуальной машиной:

Также должны учесть бинарный характер переменных xim, которые указывают, где находится каждая виртуальная машина. Для этого мы можем добавить следующие ограничения:

Недостатком данной модели является то, что она не учитывает запросы пользователей на обработку, связанные с необходимой временной задержкой с получением результатов, объемы хранимых и обрабатываемых этими приложениями данных, производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных.

**Выводы по разделу 1**

В данном разделе были рассмотрены существующие решения, направленные на оптимизацию распределенного хранения и распределенной обработки данных. Были подробно описаны для каждой модели целевая функция и ее ограничения, а также их недостатки.

# **2. ОБОСНОВАНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ СМЕШАННОГО ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ ОПТИМИЗАЦИИ РЕШЕНИЙ ПО РАСПРЕДЕЛЕННЫМ ХРАНЕНИЯМ ДАННЫХ И РАСПРЕДЕЛЕННОЙ ИХ ОБРАБОТКИ**

## **2.1 Обоснование математической модели смешанного целочисленного линейного программирования оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения операций и ограничениях на директивные сроки обработки**

С целью синтеза математической модели, смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных в рассмотрение введены следующие обозначения для индексов:

1) i - индекс типа заданий, обрабатываемых на приборе, I — количество типов заданий;

2) 1 - индекс устройства хранения данных, L — количество устройств хранения данных;

3) m - индекс устройства обработки данных, М — количество устройств обработки данных;

В рассмотрение введены обозначения для параметров, характеризующих сиcтему, реализующую распределенное хранение и обработку данных:

1) — вычислительная производительность (операций/сек) и-го устройства обработки данных (m = , вектор Р вычислительных производительностей устройств обработки данных Р = (,,..,);

2) - стоимость единицы времени выполнения вычислений на m-м вычислительном устройстве; вектор С стоимостей единиц времени выполнения вычислений на m-x вычислительных устройств С = (с­1, с2,…, см)

3) - доступный объем (байт) l-го хранилища данных (l = ), вектор V доступных размеров хранилищ данных V = (, ,…, );

4) — стоимость хранения единицы данных на l-м устройстве хранения, век-

тор Q стоимостей хранения единицы данных на l-х устройствах хранения Q =

(, ,…, );

5) — стоимость единицы времени использования канала передачи данных

между l-м устройством хранения (l = ) и m-м устройством обработки

(m = ), матрица W = , стоимостей единицы времени использования канала передачи данных между l-ми устройствами хранения (l = ) и m-ми устройствами обработки (m = ).

В рассмотрение введены обозначения параметров, характеризующих приложения, реализующие распределенную обработку данных, объемы хранимых и обрабатываемых этими приложениями данных, а также запросы пользователей на обработку, связанные необходимой временной задержкой с получением результатов

1) - объем данных i-го типа, распределенное хранение и обработка которых реализуется в системе, вектор V количества данных i-х приложений, которые будут обрабатываться в системе: V = (,,..., );

2) - объем вычислений (количество вычислительных операций), которые требуется выполнить при обработке данных i-го типа, вектора S количества вычислительных операций, которые необходимо выполнить при обработки данных каждого i-го типа: S = ()

3) — временной интервал, по истечение которого пользователю требуется получить результаты обработки данных i-го типа (временная задержка с получением результатов обработки данных i-го типа), вектор D значений временных задержек с получением результатов обработки данных i-x типов D = ())

В рассмотрение введены обозначения для целочисленных и бинарных пере­менных, значения которых соответствуют решению рассматриваемой задачи:

1) - вычислительная производительность (операций/сек), которая выделя­ется для обработки данных i-го типа на m-м устройстве, матрица - матрица значений вычислительных производительностей, выделенных на m-х устройствах для обработки данных i-х типов (i = ; m = );

2) - переменная, принимающая значение 1 в случае, если хранение дан­ных i-гo типа назначено на l-e устройство; матрица назначений дан­ных i-x типов (i = ) для их хранения на 1-х (l = ) устройствах;

3) *-* переменная, принимающая значение 1 в случае, если обработка дан­ных i-гo типа назначена на m-e устройство; матрица назначений данных i-x типов (i = ) для их обработки на m-e (l = ) устройства.

В рассматриваемой системе ограниченными ресурсами устройств, реализую­щих распределенное хранение, обработку и передачу данных, являются: 1) объем хранилищ данных (общее количество данных, которое может быть размещено в хранилищах); 2) производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных. Тогда на каждом из устройств хранения может размещаться данных не более, чем его объем. В этом случае ограничение на количество хранимых на устройствах данных имеет вид:

(2.1)

Суммарная вычислительная производительность, назначаемая приложениям, реализующим на m-м устройстве обработку данных i-x типов, не превышает общую вычислительную производительность этого m-гo устройства. Тогда ограниче­ние на используемые вычислительные ресурсы (производительность вычислитель­ного устройства), выделяемые на m-м устройстве приложениям для обработки ими данных, имеет следующий вид:

(2.2)

при m =

Наряду с ограничениями на выделяемые для хранения, обработки и передачи данных ресурсы требуется ограничить интервалы времени, в течение которых будет реализовываться передача данных из хранилищ, а также их обработка. То есть необходимо ограничить интервалы времени, по истечение которых пользователями ресурсов будут получены результаты обработки данных. Время на передачу всего объема данных i-гo типа, обозначенного через , по каналу свази между l-m устройством хранение (где они размешены) и m-м устройством обработки в соответствии с решением, характеризуемым переменными , , с учетом производительности канала передачи данных между этими устройствами определяется выражением вида:

(2.3)

Если - объём вычислений (количество операций), которые должны быть выполнены для обработки данных i-гo типа, а - это производительность вычислений (операций/сек), назначенная для обработки данных i-гo типа на m-м устройстве, тогда длительность выполнения операций по обработке данных i-гo типа на назначенном для этого m-м устройстве определяется выражением вида:

(2.4)

В силу того, что в рассмотрение введено время задержки на выполнение опе­раций с данными i-гo типа, обозначенное через тогда назначение устройств для их хранения и обработки, а также определение производительности m-го вычислительного устройства, выделяемого для обработки, и производительности канала передачи данных между l-м устройством хранения и m-м устройством обработки реализуется в соответствии с ограничением следующего вида:

(2.5)

,

где i = .

При условии, что выражение вида

позволяет идентифицировать количество данных i-x типов, хранение которых закреплено на некоторым l-м устройством, а через 𝑞𝑙 обозначена стоимость хранения единицы данных на этих устройствах, тогда определение общей (суммарной) стоимости хранения всех данных l типов на всех устройствах (в количестве L) определяется выражением следующего вида:

(2.6)

Длительность выполнения операций обработки данных i-х типов на m-м вычислительном устройстве (при учете выделенных для этого производительностей рассматриваемого m-го ВУ) определяется выражением вида

(2.7)

Стоимость единицы времени реализации вычислений обозначена через , тогда выражение, позволяющее определить общую стоимость обработки данных всех I типов на m-x устройствах, определяется выражением вида:

C учетом того, что длительность интервала времени передачи данных i-го ти­па между l-м устройством, на котором назначено их хранение, и m-м устройством, на котором назначена их обработка (при учете производительности канала, с использованием которого реализуется передача данных i-гo типа) определяется выражением (3). а стоимость единицы времени использования канала передачи данных между этими устройствами обозначена через . тогда стоимость переда­чи данных всех / типов будет определяться выражением вида:

(2.8)

Полученные выражения (6), (7), (8) используются при синтезе критерия оптимизации решений, представляющую собой суммарную стоимость хранения, обработки и передачи данных между устройствами. В дополнение к условиям (1). (2). (5), являющимися ограничениями на множество допустимых решений задачи, введено ограничение, определяющее возможность использования только одного устройства хранения и одного устройства обработки для выполнения соответствующих операций с рассматриваемыми данными i-го типа. В результате математическая модель задачи смешанного целочисленного программирования получена в следующем виде:

1) критерий оптимизации решений по распределенному хранению и обработке данных:

min f (; ; ),

(2.9)

где f (; ; ) =

2) ограничения на множество допустимых решений:

* ограничения на возможность хранения данных i-го типа на одном l-м устройстве и обработки данных на одном т-м устройстве:
* ограничение на объемы хранимых на устройстве данных:
* ограничение на количество выделенной вычислительной производительности на m-м устройстве для обработки данных i-х типов:
* ограничение на длительность выполнения операций по передаче данных между устройствами хранения и обработки и выполнения вычислений с данными i-го ти­па:

Критерий (10) является нелинейным. С целью его линеаризации в рассмотре­ние введена дополнительная переменная , которая представляет собой произве­дение переменных и : = : . Корректность замены произведения \* на переменную обеспечивается введением следующих дополнительных условий: и . После введения дополнительной переменной математическая модель MILP оптимизации решений по распределенному хранению и обработке имеет следующий итоговый вид:

1) критерий оптимизации решений по распределенному хранению и обработке данных:

min f (; ; ), где

f (; ; ) ==

(2.10)

2) ограничения на множество допустимых решений:

а)

(2.11)

, ,

б)

(2.12)

в)

(2.13)

г)

(2.14)

д)

(2.15)

е)

(2.16)

Ограничение (е) искусственное, если , то

Полученные выражения (2.11) - (2.16) представляют модель MILP оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения указанных операций и ограничениях на длительности временных задержек на передачу и обработку данных, задаваемых пользователем.

## **2.2 Обоснование математической модели смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения**

С целью синтеза математической модели, смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных в рассмотрение введены следующие обозначения для индексов:

1) i - индекс типа заданий, обрабатываемых на приборе, I — количество типов заданий;

2) 1 - индекс устройства хранения данных, L — количество устройств хранения данных;

3) m - индекс устройства обработки данных, М — количество устройств обработки данных;

В рассмотрение введены обозначения для параметров, характеризующих сиcтему, реализующую распределенное хранение и обработку данных:

1) — вычислительная производительность (операций/сек) и-го устройства обработки данных (m = , вектор Р вычислительных производительностей устройств обработки данных Р = (,,..,);

2) - стоимость единицы времени выполнения вычислений на m-м вычислительном устройстве; вектор С стоимостей единиц времени выполнения вычислений на m-x вычислительных устройств С = (с­1, с2, … , см)

3) - доступный объем (байт) l-го хранилища данных (l = ), вектор V доступных размеров хранилищ данных V = (, ,…, );

4) — стоимость хранения единицы данных на l-м устройстве хранения, вектор Q стоимостей хранения единицы данных на l-х устройствах хранения Q = (, ,…, );

5) — стоимость единицы времени использования канала передачи данных между l-м устройством хранения (l = ) и m-м устройством обработки (m = ), матрица W = , стоимостей единицы времени использования канала передачи данных между l-ми устройствами хранения (l = ) и m-ми устройствами обработки (m = ).

В рассмотрение введены обозначения параметров, характеризующих приложения, реализующие распределенную обработку данных, объемы хранимых и обрабатываемых этими приложениями данных, а также запросы пользователей на обработку, связанные необходимой временной задержкой с получением результатов

1) - объем данных i-го типа, распределенное хранение и обработка которых

реализуется в системе, вектор V количества данных i-х приложений, которые будут обрабатываться в системе: V = (,,..., );

2) - объем вычислений (количество вычислительных операций), которые требуется выполнить при обработке данных i-го типа, вектора S количества вычислительных операций, которые необходимо выполнить при обработки данных каждого i-го типа: S = ().

В рассмотрение введены обозначения для целочисленных и бинарных переменных, значения которых соответствуют решению рассматриваемой задачи:

1) - вычислительная производительность (операций/сек), которая выделя­ется для обработки данных i-го типа на m-м устройстве, матрица - матрица значений вычислительных производительностей, выделенных на m-х устройствах для обработки данных i-х типов (i = ; m = );

2) - переменная, принимающая значение 1 в случае, если хранение данных i-гo типа назначено на l-e устройство; матрица назначений дан­ных i-x типов (i = ) для их хранения на 1-х (l = ) устройствах;

3) *-* переменная, принимающая значение 1 в случае, если обработка дан­ных i-гo типа назначена на m-e устройство; матрица назначений данных i-x типов (i = ) для их обработки на m-e (l = ) устройства.

В рассматриваемой системе ограниченными ресурсами устройств, реализующих распределенное хранение, обработку и передачу данных, являются: 1) объем хранилищ данных (общее количество данных, которое может быть размещено в хранилищах); 2) производительность вычислительных устройств, которая разделяется между приложениями, реализующими обработку данных. Тогда на каждом из устройств хранения может размещаться данных не более, чем его объем. В этом случае ограничение на количество хранимых на устройствах данных имеет вид:

Суммарная вычислительная производительность, назначаемая приложениям, реализующим на m-м устройстве обработку данных i-x типов, не превышает об­щую вычислительную производительность этого m-гo устройства. Тогда ограничение на используемые вычислительные ресурсы (производительность вычислительного устройства), выделяемые на m-м устройстве приложениям для обработки ими данных, имеет следующий вид:

при m =

Если - объём вычислений (количество операций), которые должны быть выполнены для обработки данных i-гo типа, а - это производительность вычислений (операций/сек), назначенная для обработки данных i-гo типа на m-м устройстве, тогда длительность выполнения операций по обработке данных i-гo типа на назначенном для этого m-м устройстве определяется выражением вида:

Стоимость единицы времени реализации вычислений обозначена через , тогда выражение, позволяющее определить общую стоимость обработки данных всех I типов на m-x устройствах, определяется выражением вида:

C учетом того, что длительность интервала времени передачи данных i-го типа между l-м устройством, на котором назначено их хранение, и m-м устройством, на котором назначена их обработка (при учете производительности канала, с использованием которого реализуется передача данных i-гo типа) определяется выражением (3). а стоимость единицы времени использования канала передачи данных между этими устройствами обозначена через . тогда стоимость переда­чи данных всех / типов будет определяться выражением вида:

(2.17)

* ограничения на возможность хранения данных i-го типа на одном l-м устройстве и обработки данных на одном т-м устройстве:
* ограничение на объемы хранимых на устройстве данных:
* ограничение на количество выделенной вычислительной производительности на m-м устройстве для обработки данных i-х типов:

Критерий (20) является нелинейным. С целью его линеаризации в рассмотре­ние введена дополнительная переменная , которая представляет собой произве­дение переменных и : = : . Корректность замены произведения \* на переменную обеспечивается введением следующих дополнительных условий: и . После введения дополнительной переменной математическая модель MILP оптимизации решений по распределенному хранению и обработке имеет следующий итоговый вид:

1. критерий оптимизации решений:

(2.18)

где

2) ограничения на множество допустимых решений:

а)

(2.19)

, -1;

б)

(2.20)

в)

(2.23)

г)

(2.24)

д)

(2.25)

е)

(2.26)

В итоге выражения (2.18) - (2.26) представляют собой модель MILP оптимизации решений по распределенному хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения.

**Выводы по разделу 2**

В данном разделе была обоснована две математической модели смешанного линейного целочисленного программирования для оптимизации решений по распределенному хранению и обработке данных.

Первая модель основывается на минимизации суммарной стоимости выполнения операций при ограничениях на директивные сроки обработки. Модель включается в себя следующие ограничения: на количество выделенной вычислительной производительности на m-м устройстве для обработки данных i-х типов, ограничение на объемы хранимых на устройстве данных, ограничения на возможность хранения данных i-го типа на одном l-м устройстве и обработки данных на одном т-м устройстве и.т.д.

Вторая модель основывается на минимизации времени на реализацию операций. Модель включается в себя следующие ограничения: на количество хранимых на устройствах данных, ограниче­ние на используемые вычислительные ресурсы (производительность вычислительного устройства), выделяемые на m-м устройстве приложениям для обработки ими данных и.т.д

# **3. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ РЕШЕНИЯ ЗАДАЧ СМЕШАННОГО ЦЕЛОЧИСЛЕННОГО ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ**

## **3.1. Выбор программного обеспечения**

Для выбора программного обеспечения, способного эффективно решать задачи смешанного целочисленного линейного программирования (Mixed Integer Linear Programming) в данной работе применен метод анализа иерархий.

Задачи MILP представляют собой математические модели, в которых некоторые переменные могут принимать только целочисленные значения, в то время как другие переменные допускают дробные значения. Такие задачи возникают в различных областях, включая логистику, производство, транспортировку, финансы и другие.

Выбор подходящего программного обеспечения для MILP задач является сложной задачей, поскольку требуется учитывать множество факторов, таких как точность решения, скорость вычислений, использование ресурсов памяти и другие. В данном контексте, метод анализа иерархий (МАИ) представляет собой полезный инструмент для сравнительного анализа и выбора программного обеспечения.

Метод анализа иерархий разработан Т.Л. Саати и представляет собой структурированный подход к принятию решений, основанный на иерархической модели предпочтений. В процессе применения МАИ определяются критерии, влияющие на выбор программного обеспечения, и оценивается их относительная важность. Затем проводится попарное сравнение программного обеспечения в соответствии с каждым критерием. Далее, используя математические методы, такие как собственные значения и векторы, определяется совокупная оценка каждого программного обеспечения и ранжирование по предпочтительности.

При этом особое внимание будет уделено определению соответствующих критериев, влияющих на эффективность решения MILP задач. Используя методы МАИ, будет определена наиболее предпочтительная альтернатива, то есть программное обеспечение, которое наиболее эффективно справляется с MILP задачами.

Результаты данного исследования позволят выбрать подходящее программное обеспечение для MILP задач на основе анализа иерархий. Это имеет практическую значимость, так как правильный выбор программного обеспечения может существенно повлиять на эффективность решения задач, связанных с оптимизацией и планированием.

Использование критериев в процессе применения метода анализа иерархий позволит сравнить и ранжировать различные программные обеспечения MILP задач. Таким образом, будет возможно выбрать оптимальное программное обеспечения, который обеспечит баланс между высокой точностью и эффективной производительностью в контексте решения конкретных MILP задач.

Иерархическая структура для выбора программного обеспечения MILP задач будет состоять из трех уровней: цель, критерии и альтернативы.

На верхнем уровне иерархии находится цель "Выбор программного обеспечения". Цель определяет общую задачу выбора оптимального программного обеспечения для решения MILP задач.

На следующем уровне расположены шесть критерий: "Производительность", "Точность", “Надежность”, “Удобность’, “Совместимость” и “Стоимость”. Эти критерии являются подкритериями, которые помогут оценить программное обеспечение на основе их способности обеспечить требуемую производительность, точность и т.д.

На самом нижнем уровне иерархии находятся альтернативы или программные обеспечения, которые будут сравниваться и оцениваться. Каждое программное обеспечение представляет собой отдельную альтернативу, которая может быть выбрана для решения MILP задач. Примерами альтернатив могут быть известные программные обеспечения MILP, такие как CPLEX, SCIP, Or-tools. Иерархическая структура представлена на рисунке 1.

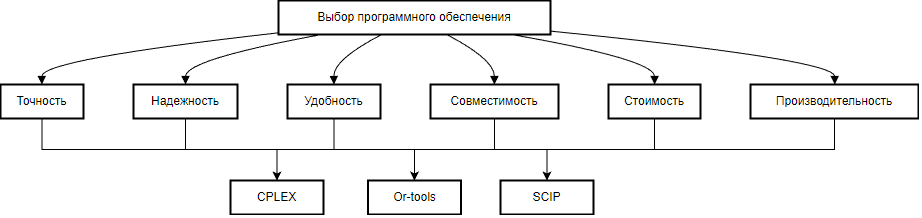


Рисунок 1 – Иерархическая структура для выбора программного обеспечения

В процессе применения метода анализа иерархий, будет проведено попарное сравнение программных обеспечений по каждому из критериев.

Таким образом, иерархическая структура, основанная на цели, критериях и альтернативах, позволит систематически и объективно выбрать наиболее подходящее программное обеспечения для решения MILP задач.

Матрица парных сравнений А1 (важность характеристики относительно цели) следующий:

Получены следующие значения элементов вектора приоритетов (j = ): = (0.32;0.14;0.03;0.13;0.24;0.14).

В этом случае вес = 0.32 соответствует точности в общей цели выбора программного обеспечения, вес = 0.14 – надежности в общей цели, вес – удобности в общей цели и т.д.

Тогда собственное значение матрицы A1, определяющее согласованность сформированных суждений определено равным 7.49 ( = 7.49), а индекс согласованности суждений определен равным 0.3 (ИС = 0,3)

Для уровня иерархии видов оборудования сформированы матрицы парных сравнений по каждой из характеристик . Значения элементов этих матриц определяют степень эффективности какого-либо i − го типа оборудования по сравнению с другими (l-ми) типами для соответствующей j − ой характеристики (критерия).

Вид матриц и вид собственных векторов соответствующих матриц, коэффициенты согласованности имеют следующий вид:

Точность:

*ИС = 0.025;*

Надежность:

*ИС = 0;*

Удобность:

*ИС = 0;*

Совместимость:

*ИС = 0.105;*

Стоимость:

*ИС = 0;*

Производительность:

*ИС = 0.025;*

В соответствии с полученными векторами для каждого программного обеспечения формируется свой вектор значений всех характеристик следующим образом :

Так как вектора и () сформированы, то может быть получена оценка каждого по формуле:

Полученные значения следующие: 0.37; 0.38; = 0.25. Отсюда следует, что программное обеспечение OR-TOOLS наиболее предпочтителен.

## **3.2 Описание функций, используемых в приложении Or-toolls, и их атрибутов используемых при решении задач**

Функция CreateSolver:

* Описание: создаёт линейный решатель для оптимизационных задач.
* Входные параметры: Нет.
* Возвращаемое значение: Линейный решатель.

solver = pywraplp.Solver.CreateSolver('GLOP')

Функция NumVar:

* Описание: создаёт числовую переменную для оптимизационной модели.
* Входные параметры:
* Нижняя граница переменной.
* Верхняя граница переменной.
* Имя переменной.
* Возвращаемое значение: Созданная числовая переменная.

solver = pywraplp.Solver.CreateSolver('GLOP')

x\_s[l, i] = solver.NumVar(0, 1, f'x\_s[{l},{i}]')

x\_c[l, i] = solver.NumVar(0, 1, f'x\_c[{l},{i}]')

x\_i[l, m, i] = solver.NumVar(0, 1, f'x\_i[{l},{m},{i}]')

Класс Objective:

* Описание: представляет целевую функцию для оптимизационной модели.

Атрибуты:

* SetCoefficient: задаёт коэффициент для переменной в целевой функции.
* SetMinimization: устанавливает целевую функцию для минимизации.

objective = solver.Objective()

objective.SetCoefficient(x\_s[l, i], q\_l[l] \* v\_i[i])

objective.SetCoefficient(x\_c[m, i], c\_m[m] \* s\_i[i] \* p\_im[i][m])

objective.SetCoefficient(x\_i[l, m, i],w\_lm[l \* M + m] \* v\_i[i] \* b\_lm[l][m])

objective.SetMinimization()

Класс Constraint:

* Описание: представляет ограничение для оптимизационной модели.

Атрибуты:

* SetCoefficient: задаёт коэффициент для переменной в ограничении.
* SetUb: устанавливает верхнюю границу для ограничения.

ct1 = solver.Constraint(-solver.infinity(), 0)  
ct1.SetCoefficient(x\_i[l, m, i], 1)  
ct1.SetCoefficient(x\_s[l, i], -1)

ct2 = solver.Constraint(-solver.infinity(), 0)  
ct2.SetCoefficient(x\_i[l, m, i], 1)  
ct2.SetCoefficient(x\_c[m, i], -1)

ct3 = solver.Constraint(-solver.infinity(), 0)  
ct3.SetCoefficient(x\_i[l, m, i], -1)  
ct3.SetCoefficient(x\_s[l, i], 1)  
ct3.SetCoefficient(x\_c[m, i], 1)  
ct3.SetUb(1)

ct4 = solver.Constraint(1, 1)  
ct4.SetCoefficient(x\_s[l, i], 1)

ct5 = solver.Constraint(1, 1)  
ct5.SetCoefficient(x\_c[l, i], 1)

ct6 = solver.Constraint(-solver.infinity(), V\_l[l])  
ct6.SetCoefficient(x\_s[l, i], v\_i[i])

ct7 = solver.Constraint(-solver.infinity(), P\_m[m])  
ct7.SetCoefficient(x\_c[m, i], p\_im[i][m])

ct8 = solver.Constraint(-solver.infinity(), d\_i[i])  
ct8.SetCoefficient(x\_i[l, m, i], v\_i[i] \* b\_lm[l][m])

ct8.SetCoefficient(x\_c[m, i], s\_i[i] \* p\_im[i][m])

Функция Solve:

* Описание: решает оптимизационную модель.
* Входные параметры: Нет.
* Возвращаемое значение: Статус решения (оптимальное или нет).

Атрибут solution\_value:

* Описание: Значение переменной после решения оптимизационной модели.

status = solver.Solve()

x\_il\_s.append(int(x\_s[l, i].solution\_value()))

x\_mi\_c.append(int(x\_c[m, i].solution\_value()))

p\_im\_c.append(int(p\_im[i][m] \* x\_c[m, i].solution\_value()))

Функции print:

* Описание: выводит информацию на консоль.
* Входные параметры: Строка или значения переменных для вывода.

Функции и атрибуты, описанные выше, используются в программном обеспечении OR-tools для решения оптимизационной задачи с использованием линейного программирования. Определяются переменные, коэффициенты, целевая функция и ограничения, а затем решается модель, выводится оптимальное значение и значения переменных после решения:

print ('Оптимальное значение:', solver. Objective (). Value ())  
print ('Данная целевая функция с ограничениями не имеет оптимального решения')

## **3.3 Описание синтаксических конструкций**

1. Импорт библиотеки: from ortools.linear\_solver import pywraplp - данная конструкция импортирует модуль pywraplp из библиотеки ortools.linear\_solver. Это позволяет использовать функции и классы, предоставляемые этим модулем, для решения линейных задач оптимизации.

2. Создание решателя: solver = pywraplp.Solver.CreateSolver('GLOP') - данная конструкция создает экземпляр решателя с помощью функции CreateSolver из модуля pywraplp. В данном случае используется решатель GLOP, который представляет собой оптимизационный пакет Google для линейного программирования.

3. Определение переменных: = {}, = {}, = {} - в данной конструкции создаются пустые словари , и , которые будут содержать переменные оптимизационной модели.

4. Использование циклов for для определения переменных: вложенные циклы for используются для итерации по значениям переменных l, i и m и создания соответствующих переменных в модели с помощью функции solver.NumVar. Это позволяет определить переменные , и внутри заданных диапазонов и с заданными именами.

5. Определение коэффициентов: списки , , , , , , , и матрицы , используются для задания коэффициентов целевой функции и ограничений модели. Эти коэффициенты могут быть настроены в соответствии с конкретной оптимизационной задачей.

6. Определение целевой функции: конструкция objective = solver.Objective() создает объект целевой функции с помощью метода Objective решателя. Затем используются методы SetCoefficient для установки коэффициентов целевой функции для каждой переменной.

7. Определение ограничений: используются вложенные циклы for для создания ограничений с помощью метода Constraint решателя. Метод SetCoefficient устанавливает коэффициенты переменных в ограничениях.

8. Решение модели: метод solver.Solve() вызывается для решения оптимизационной модели. Результат выполнения сохраняется в переменной.

9. Конструкция variable.solution\_value() используется для получения значения переменной в решении линейной модели.

10. Конструкция solver.Objective().Value() используется для получения оптимального значения целевой функции:

print('Оптимальное значение:', solver.Objective().Value())

В данном примере, solver.Objective().Value() возвращает численное значение оптимальной целевой функции, и оно выводится на экран с помощью функции print(). Это позволяет получить информацию о достигнутом оптимальном значении целевой функции.

## **3.4 Описание программы**

Данный код реализует решатель линейной оптимизационной задачи с использованием библиотеки OR-Tools. Решается задача оптимизации смешанного целочисленного программирования.

Давайте рассмотрим код поэтапно и опишем каждый его элемент:

1. Импорт необходимых модулей:

from ortools.linear\_solver import pywraplp

Данный код импортирует модуль pywraplp из библиотеки OR-Tools, который содержит классы и методы для работы с линейными оптимизационными задачами.

2. Создание решателя:

solver = pywraplp.Solver.CreateSolver('GLOP')

Здесь создается объект solver, который представляет собой решатель линейной оптимизационной задачи с помощью GLOP (Google Linear Optimization Package). GLOP является одним из алгоритмов решения задачи линейного программирования.

3. Определение переменных:

L = 5

I = 5

M = 5

x\_s = {}

x\_c = {}

x\_i = {}

for l in range(L):

for i in range(I):

x\_s[l, i] = solver.NumVar(0, 1, f'x\_s[{l},{i}]')

x\_c[l, i] = solver.NumVar(0, 1, f'x\_c[{l},{i}]')

for m in range(M):

x\_i[l, m, i] = solver.NumVar(0, 1, f'x\_i[{l},{m},{i}]')

В этом блоке кода определяются переменные модели. В данном случае используются три словаря x\_s, x\_c и x\_i для хранения переменных x\_s, x\_c и x\_i соответственно. Значения переменных ограничены от 0 до 1.

4. Определение коэффициентов:

q\_l = [5] \* L

v\_i = [200]

c\_m = [5] \* M

s\_i = [100]

w\_lm = [5] \* L \* M

b\_lm = [[1] \* M for \_ in range(L)]

p\_im = [[1] \* M for \_ in range(I)]

V\_l = [5] \* L

P\_m = 150,200,250,100,300

d\_i = [50]

Здесь определяются коэффициенты модели. Коэффициенты q\_l, c\_m, s\_i, w\_lm, V\_l, P\_m и d\_i, v\_i представляют собой списки с предопределенными значениями.

5. Определение целевой функции:

objective = solver.Objective()

for l in range(L):

for i in range(I):

objective.SetCoefficient(x\_s[l, i], q\_l[l] \* v\_i[i])

for m in range(M):

for i in range(I):

objective.SetCoefficient(x\_c[m, i], c\_m[m] \* s\_i[i] \* p\_im[i][m])

for i in range(I):

for m in range(M):

for l in range(L):

objective.SetCoefficient(x\_i[l, m, i],w\_lm[l \* M + m] \* v\_i[i] \* b\_lm[l][m])

objective.SetMinimization()

Здесь создается объект objective, который представляет целевую функцию оптимизационной задачи. Для каждой переменной x\_s, x\_c и x\_i устанавливаются коэффициенты, определенные на предыдущем этапе. Целевая функция устанавливается на минимум.

6. Определение ограничений:

for l in range(L):

for i in range(I):

for m in range(M):

ct1 = solver.Constraint(-solver.infinity(), 0)

ct1.SetCoefficient(x\_i[l, m, i], 1)

ct1.SetCoefficient(x\_s[l, i], -1)

ct2 = solver.Constraint(-solver.infinity(), 0)

ct2.SetCoefficient(x\_i[l, m, i], 1)

ct2.SetCoefficient(x\_c[m, i], -1)

ct3 = solver.Constraint(-solver.infinity(), 0)

ct3.SetCoefficient(x\_i[l, m, i], -1)

ct3.SetCoefficient(x\_s[l, i], 1)

ct3.SetCoefficient(x\_c[m, i], 1)

ct3.SetUb(1)

for i in range(I):

ct4 = solver.Constraint(1, 1)

for l in range(L):

ct4.SetCoefficient(x\_s[l, i], 1)

for i in range(I):

ct5 = solver.Constraint(1, 1)

for m in range(M):

ct5.SetCoefficient(x\_c[l, i], 1)

for l in range(L):

ct6 = solver.Constraint(-solver.infinity(), V\_l[l])

for i in range(I):

ct6.SetCoefficient(x\_s[l, i], v\_i[i])

for m in range(M):

ct7 = solver.Constraint(-solver.infinity(), P\_m[m])

for i in range(I):

ct7.SetCoefficient(x\_c[m, i], p\_im[i][m])

for i in range(I):

ct8 = solver.Constraint(-solver.infinity(), d\_i[i])

for l in range(L):

for m in range(M):

ct8.SetCoefficient(x\_i[l, m, i], v\_i[i] \* b\_lm[l][m])

for m in range(M):

ct8.SetCoefficient(x\_c[m, i], s\_i[i] \* p\_im[i][m])

В этом блоке кода определяются ограничения для переменных модели. Создаются объекты типа Constraint (ограничение) с заданными верхними и нижними границами. Затем устанавливаются коэффициенты для каждого ограничения.

7. Решение модели и вывод результатов:

status = solver.Solve()

if status == pywraplp.Solver.OPTIMAL:

print('Оптимальное значение:', solver.Objective().Value())

else:

print('Данная целевая функция с ограничениями не имеет оптимального решения')

print('Решение для x\_il^s:')

for l in range(L):

x\_il\_s = []

for i in range(I):

x\_il\_s.append(int(x\_s[l, i].solution\_value()))

print(x\_il\_s)

print('Решение для x\_mi^c:')

for m in range(M):

x\_mi\_c = []

for i in range(I):

x\_mi\_c.append(int(x\_c[m, i].solution\_value()))

print(x\_mi\_c)

print('Решение для p\_im^c:')

for i in range(I):

p\_im\_c = []

for m in range(M):

p\_im\_c.append(int(p\_im[i][m] \* x\_c[m, i].solution\_value()))

print(p\_im\_c)

В этой части кода решается модель с заданными ограничениями и целевой функцией. Затем проверяется статус решения и выводится оптимальное значение целевой функции, если решение найдено. В противном случае выводится сообщение об отсутствии оптимального решения.

Далее выводятся решения для переменных x\_s, x\_c и p\_im. Значения переменных извлекаются с помощью метода solution\_value(). Значения x\_s и x\_c преобразуются в целочисленные значения с помощью функции int(). Значения p\_im вычисляются путем умножения p\_im[i][m] на x\_c[m, i].

**Выводы по разделу 3**

В данном разделе был рассмотрен метод анализа иерархии для выбора программного обеспечения для распределённого хранения и обработки данных.

Для решения задачи выбора программного обеспечения был использован метод анализа иерархии, который позволяет определить относительную важность критериев и альтернатив при выборе программного обеспечения на основе экспертных оценок.

Для решения задачи оптимизации была использована библиотека or-tools, которая предоставляет набор функций для решения задач линейного и целочисленного программирования, в том числе и задач оптимизации смешанного целочисленного программирования.

Таким образом, использование метода анализа иерархии и библиотеки or-tools позволяет эффективно выбирать программное обеспечение для распределенного хранения и обработки данных и решать задачи оптимизации в этой области

# **4. АНАЛИЗ ЭФФЕКТИВНОСТИ ПРИМЕНЕНИЯ МОДЕЛЕЙ, НАПРАВЛЕННЫХ НА ОПТИМИЗАЦИЮ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ И РАСПРЕДЕЛЕННОЙ ОБРАБОТКИ ДАННЫХ**

## **4.1. Анализ первой математической модели смешанного целочисленного линейного программирования оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения операций и ограничениях на директивные сроки обработки.**

При значении используются следующие параметры: У параметра значения, следующие : 100, 200, 300, 400, 500, 600. У параметра значения, следующие: 100, 200, 300, 400, 500, 600. У параметра значения, следующие: 50, 80, 100, 150. У параметра значения, следующие: 10, 20, 30, 40, 50, 60. Данные параметры используется в модели при поиске решений. Результат решения задачи по оптимизации решений по распределённому хранению и обработке данных при минимизации суммарной стоимости выполнения операций и ограничениях на директивные сроки соединены в таблицу.

При L = 5, M = 5, I = 5 выведем следующие результаты:

Таблица 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Оптимальное решение целевой функции |
| 100 | 200 | 50 | 10 | 45 |
|  |  |  | 20 | 50 |
| 30 | 55 |
| 40 | 60 |
| 50 | 65 |
| 60 | 70 |
| 80 | 10 | 55 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 10 | 45 |
| 20 | 60 |
| 30 | 65 |
| 40 | 70 |
| 50 | 75 |
| 60 | 80 |
| 100 | 10 | 65 |
|  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 150 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 100 | 250 | 50 | 10 | 55 |
|  |  |  | 20 | 60 |
| 30 | 65 |
| 40 | 70 |
| 50 | 75 |
| 60 | 80 |
| 80 | 10 | 65 |
|  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 100 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 150 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 100 | 300 | 50 | 10 | 65 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 80 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 100 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 150 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 350 | 50 | 10 | 75 |
|  |  |  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 80 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 100 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 150 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 100 | 400 | 50 | 10 | 85 |
|  |  |  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 80 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 150 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 450 | 50 | 10 | 95 |
|  |  |  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 80 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 100 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 140 |
| 150 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 200 | 200 | 50 | 10 | 60 |
|  |  |  | 20 | 65 |
| 30 | 70 |
| 40 | 75 |
| 50 | 80 |
| 60 | 85 |
| 80 | 10 | 70 |
|  | 20 | 75 |
| 30 | 80 |
| 40 | 85 |
| 50 | 90 |
| 60 | 95 |
| 100 | 10 | 80 |
|  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 150 | 10 | 90 |
|  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 200 | 250 | 50 | 10 | 70 |
|  |  |  | 20 | 75 |
| 30 | 80 |
| 40 | 85 |
| 50 | 90 |
| 60 | 95 |
| 80 | 10 | 80 |
|  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 100 | 10 | 90 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 150 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 200 | 300 | 50 | 10 | 80 |
|  |  |  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 80 | 10 | 90 |
|  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 100 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 150 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 200 | 350 | 50 | 10 | 90 |
|  |  |  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 80 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 100 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 150 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 200 | 400 | 50 | 10 | 100 |
|  |  |  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 80 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 100 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 150 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 200 | 450 | 50 | 10 | 110 |
|  |  |  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 135 |
| 80 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 100 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 150 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 300 | 200 | 50 | 10 | 75 |
|  |  |  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 80 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 100 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 150 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 300 | 250 | 50 | 10 | 85 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 80 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 150 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 300 | 300 | 50 | 10 | 95 |
|  |  |  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 80 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 100 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 150 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 300 | 350 | 50 | 10 | 105 |
|  |  |  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 80 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 150 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 300 | 400 | 50 | 10 | 115 |
|  |  |  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 80 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 100 | 10 | 135 |
| 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 160 |
| 150 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 300 | 450 | 50 | 10 | 125 |
|  |  |  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 80 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 100 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 150 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 400 | 200 | 50 | 10 | 90 |
|  |  |  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 80 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 100 | 10 | 110 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 150 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 400 | 250 | 50 | 10 | 100 |
|  |  |  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 80 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 100 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 150 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 400 | 300 | 50 | 10 | 110 |
|  |  |  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 80 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 100 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 150 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 400 | 350 | 50 | 10 | 120 |
|  | |  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 80 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 100 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 150 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 400 | 400 | 50 | 10 | 130 |
|  |  |  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 155 |
| 80 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 100 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 150 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 400 | 450 | 50 | 10 | 140 |
|  |  |  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 80 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 100 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 150 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 500 | 200 | 50 | 10 | 105 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 80 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 150 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 500 | 250 | 50 | 10 | 115 |
|  |  |  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 80 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 100 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 150 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 500 | 300 | 50 | 10 | 125 |
|  |  |  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 80 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 100 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 150 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 500 | 350 | 50 | 10 | 135 |
|  |  |  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 80 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 100 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 180 |
| 150 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 500 | 400 | 50 | 10 | 145 |
|  |  |  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 80 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 100 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 150 | 10 | 175 |
|  | 20 | 180 |
| 30 | 185 |
| 40 | 190 |
| 50 | 195 |
| 60 | 200 |
| 500 | 450 | 50 | 10 | 155 |
|  |  |  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 80 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 100 | 10 | 175 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 180 |
| 30 | 185 |
| 40 | 190 |
| 50 | 195 |
| 60 | 200 |
| 150 | 10 | 185 |
|  | 20 | 190 |
| 30 | 195 |
| 40 | 200 |
| 50 | 205 |
| 60 | 210 |
| 600 | 200 | 50 | 10 | 120 |
|  |  |  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 80 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 100 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 150 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 600 | 250 | 50 | 10 | 130 |
|  |  |  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 80 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 100 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 150 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 600 | 300 | 50 | 10 | 140 |
|  |  |  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 80 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 100 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 150 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 600 | 350 | 50 | 10 | 150 |
|  |  |  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 175 |
| 80 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 100 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 150 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 600 | 400 | 50 | 10 | 160 |
|  |  |  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 80 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 100 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 150 | 10 | 190 |
|  | 20 | 195 |
| 30 | 200 |
| 40 | 205 |
| 50 | 210 |
| 60 | 215 |
| 600 | 450 | 50 | 10 | 170 |

Продолжение таблицы 1. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 80 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 100 | 10 | 190 |
|  | 20 | 195 |
| 30 | 200 |
| 40 | 205 |
| 50 | 210 |
| 60 | 215 |
| 150 | 10 | 200 |
|  | 20 | 205 |
| 30 | 210 |
| 40 | 215 |
| 50 | 220 |
| 60 | 225 |

Рисунок 1 – Оптимальное решение целевой функции при Si = 100, Vi = 200

Рисунок 2 – Оптимальное решение целевой функции при Si = 100, Vi = 250

****

Рисунок 3 – Оптимальное решение целевой функции при Si = 100, Vi = 300

****

Рисунок 4 – Оптимальное решение целевой функции при Si = 100, Vi = 350

****

Рисунок 5 – Оптимальное решение целевой функции при Si = 100, Vi = 400

Рисунок 6 – Оптимальное решение целевой функции при Si = 100, Vi = 450

****

Рисунок 7 – Оптимальное решение целевой функции при Si = 200, Vi = 200

****

Рисунок 8 – Оптимальное решение целевой функции при Si = 200, Vi = 250

****

Рисунок 9 – Оптимальное решение целевой функции при Si = 200, Vi = 300

****

Рисунок 10 – Оптимальное решение целевой функции при Si = 200, Vi = 350

****

Рисунок 11 – Оптимальное решение целевой функции при Si = 200, Vi = 400

****

Рисунок 12 – Оптимальное решение целевой функции при Si = 200, Vi = 450

****

Рисунок 13 – Оптимальное решение целевой функции при Si = 300, Vi = 200

****

Рисунок 14 – Оптимальное решение целевой функции при Si = 300, Vi = 250

****

Рисунок 15 – Оптимальное решение целевой функции при Si = 300, Vi = 300

****

Рисунок 16 – Оптимальное решение целевой функции при Si = 300, Vi = 350

****

Рисунок 17 – Оптимальное решение целевой функции при Si = 300, Vi = 400

****

Рисунок 18 – Оптимальное решение целевой функции при Si = 300, Vi = 450

****

Рисунок 19 – Оптимальное решение целевой функции при Si = 400, Vi = 200

****

Рисунок 21 – Оптимальное решение целевой функции при Si = 400, Vi = 250

****

Рисунок 22 – Оптимальное решение целевой функции при Si = 400, Vi = 300

****

Рисунок 23 – Оптимальное решение целевой функции при Si = 400, Vi = 350

****

Рисунок 24 – Оптимальное решение целевой функции при Si = 400, Vi = 400

****

Рисунок 25 – Оптимальное решение целевой функции при Si = 400, Vi = 450

****

Рисунок 21 – Оптимальное решение целевой функции при Si = 500, Vi = 200

****

Рисунок 22 – Оптимальное решение целевой функции при Si = 500, Vi = 250

****

Рисунок 23 – Оптимальное решение целевой функции при Si = 500, Vi = 300

****

Рисунок 24 – Оптимальное решение целевой функции при Si = 500, Vi = 350

****

Рисунок 25 – Оптимальное решение целевой функции при Si = 500, Vi = 400

****

Рисунок 26 – Оптимальное решение целевой функции при Si = 500, Vi = 450

****

Рисунок 27 – Оптимальное решение целевой функции при Si = 600, Vi = 350

****

Рисунок 28 – Оптимальное решение целевой функции при Si = 600, Vi = 400

****

Рисунок 29 – Оптимальное решение целевой функции при Si = 600, Vi = 450

## **4.2. Анализ второй математической модели смешанного целочисленного линейного программирования оптимизации решений по распределенному хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения**

При значении используются следующие параметры: У параметра значения, следующие : 100, 200, 300, 400, 500, 600. У параметра значения, следующие: 100, 200, 300, 400, 500, 600. У параметра значения, следующие: 50, 80, 100, 150. У параметра значения, следующие: 10, 20, 30, 40, 50, 60. Данные параметры используется в модели при поиске решений. Результат решения задачи по оптимизации решений по распределённому хранению и обработке данных при минимизации времени на реализацию операций и при ограничении на стоимости их выполнения.

При L = 5, M = 5, I = 5 выведем следующие результаты:

Таблица 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | Оптимальное решение целевой функции |
| 100 | 200 | 50 | 10 | 45 |
|  |  |  | 20 | 50 |
| 30 | 55 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 60 |
| 50 | 65 |
| 60 | 70 |
| 80 | 10 | 55 |
|  | 20 | 60 |
| 30 | 65 |
| 40 | 70 |
| 50 | 75 |
| 60 | 80 |
| 100 | 10 | 65 |
|  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 150 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 100 | 250 | 50 | 10 | 55 |
|  |  |  | 20 | 60 |
| 30 | 65 |
| 40 | 70 |
| 50 | 75 |
| 60 | 80 |
| 80 | 10 | 65 |
|  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 100 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 150 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 50 | 105 |
| 60 | 110 |
| 100 | 300 | 50 | 10 | 65 |
|  |  |  | 20 | 70 |
| 30 | 75 |
| 40 | 80 |
| 50 | 85 |
| 60 | 90 |
| 80 | 10 | 75 |
|  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 100 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 150 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 350 | 50 | 10 | 75 |
|  |  |  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 80 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 100 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 120 |
| 150 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 100 | 400 | 50 | 10 | 85 |
|  |  |  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 80 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 150 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 450 | 50 | 10 | 95 |
|  |  |  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 80 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 100 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 150 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 200 | 200 | 50 | 10 | 60 |
|  |  |  | 20 | 65 |
| 30 | 70 |
| 40 | 75 |
| 50 | 80 |
| 60 | 85 |
| 80 | 10 | 70 |
|  | 20 | 75 |
| 30 | 80 |
| 40 | 85 |
| 50 | 90 |
| 60 | 95 |
| 100 | 10 | 80 |
|  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 150 | 10 | 90 |
|  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 200 | 250 | 50 | 10 | 70 |
|  |  |  | 20 | 75 |
|  |  |  | 30 | 80 |
|  |  |  | 40 | 85 |
|  |  |  | 50 | 90 |
|  |  |  | 60 | 95 |
|  |  | 80 | 10 | 80 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 100 | 10 | 90 |
|  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 150 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 200 | 300 | 50 | 10 | 80 |
|  |  |  | 20 | 85 |
| 30 | 90 |
| 40 | 95 |
| 50 | 100 |
| 60 | 105 |
| 80 | 10 | 90 |
|  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 100 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 150 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 200 | 350 | 50 | 10 | 90 |
|  |  |  | 20 | 95 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 80 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 100 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 150 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 200 | 400 | 50 | 10 | 100 |
|  |  |  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 80 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 100 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 150 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 200 | 450 | 50 | 10 | 110 |
|  |  |  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 80 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 100 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 150 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 300 | 200 | 50 | 10 | 75 |
|  |  |  | 20 | 80 |
| 30 | 85 |
| 40 | 90 |
| 50 | 95 |
| 60 | 100 |
| 80 | 10 | 85 |
|  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 100 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 50 | 115 |
| 60 | 120 |
| 150 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 300 | 250 | 50 | 10 | 85 |
|  |  |  | 20 | 90 |
| 30 | 95 |
| 40 | 100 |
| 50 | 105 |
| 60 | 110 |
| 80 | 10 | 95 |
|  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 100 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 150 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 300 | 300 | 50 | 10 | 95 |
|  |  |  | 20 | 100 |
| 30 | 105 |
| 40 | 110 |
| 50 | 115 |
| 60 | 120 |
| 80 | 10 | 105 |
|  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 130 |
| 100 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 150 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 300 | 350 | 50 | 10 | 105 |
|  |  |  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 80 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 150 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 300 | 400 | 50 | 10 | 115 |
|  |  |  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | 80 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 100 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 150 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 300 | 450 | 50 | 10 | 125 |
|  |  |  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 80 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 100 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 150 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 400 | 200 | 50 | 10 | 90 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 95 |
| 30 | 100 |
| 40 | 105 |
| 50 | 110 |
| 60 | 115 |
| 80 | 10 | 100 |
|  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 100 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 150 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 400 | 250 | 50 | 10 | 100 |
|  |  |  | 20 | 105 |
| 30 | 110 |
| 40 | 115 |
| 50 | 120 |
| 60 | 125 |
| 80 | 10 | 110 |
|  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 100 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 150 | 10 | 130 |
|  | 20 | 135 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 400 | 300 | 50 | 10 | 110 |
|  |  |  | 20 | 115 |
| 30 | 120 |
| 40 | 125 |
| 50 | 130 |
| 60 | 135 |
| 80 | 10 | 120 |
|  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 100 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 150 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 400 | 350 | 50 | 10 | 120 |
|  |  |  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 80 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 100 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 150 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 400 | 400 | 50 | 10 | 130 |
|  |  |  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 80 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 100 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 150 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 400 | 450 | 50 | 10 | 140 |
|  |  |  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 80 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 50 | 170 |
| 60 | 175 |
| 100 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 150 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 500 | 200 | 50 | 10 | 105 |
|  |  |  | 20 | 110 |
| 30 | 115 |
| 40 | 120 |
| 50 | 125 |
| 60 | 130 |
| 80 | 10 | 115 |
|  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |
| 60 | 140 |
| 100 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 150 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 500 | 250 | 50 | 10 | 115 |
|  |  |  | 20 | 120 |
| 30 | 125 |
| 40 | 130 |
| 50 | 135 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 140 |
| 80 | 10 | 125 |
|  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 100 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 150 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 500 | 300 | 50 | 10 | 125 |
|  |  |  | 20 | 130 |
| 30 | 135 |
| 40 | 140 |
| 50 | 145 |
| 60 | 150 |
| 80 | 10 | 135 |
|  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 100 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 150 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 500 | 350 | 50 | 10 | 135 |
|  |  |  | 20 | 140 |
| 30 | 145 |
| 40 | 150 |
| 50 | 155 |
| 60 | 160 |
| 80 | 10 | 145 |
|  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 100 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 150 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 500 | 400 | 50 | 10 | 145 |
|  |  |  | 20 | 150 |
| 30 | 155 |
| 40 | 160 |
| 50 | 165 |
| 60 | 170 |
| 80 | 10 | 155 |
|  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 100 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 150 | 10 | 175 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 20 | 180 |
| 30 | 185 |
| 40 | 190 |
| 50 | 195 |
| 60 | 200 |
| 500 | 450 | 50 | 10 | 155 |
|  |  |  | 20 | 160 |
| 30 | 165 |
| 40 | 170 |
| 50 | 175 |
| 60 | 180 |
| 80 | 10 | 165 |
|  | 20 | 170 |
| 30 | 175 |
| 40 | 180 |
| 50 | 185 |
| 60 | 190 |
| 100 | 10 | 175 |
|  | 20 | 180 |
| 30 | 185 |
| 40 | 190 |
| 50 | 195 |
| 60 | 200 |
| 150 | 10 | 185 |
|  | 20 | 190 |
| 30 | 195 |
| 40 | 200 |
| 50 | 205 |
| 60 | 210 |
| 600 | 200 | 50 | 10 | 120 |
|  |  |  | 20 | 125 |
| 30 | 130 |
| 40 | 135 |
| 50 | 140 |
| 60 | 145 |
| 80 | 10 | 130 |
|  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 100 | 10 | 140 |
|  | 20 | 145 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 150 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 600 | 250 | 50 | 10 | 130 |
|  |  |  | 20 | 135 |
| 30 | 140 |
| 40 | 145 |
| 50 | 150 |
| 60 | 155 |
| 80 | 10 | 140 |
|  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 100 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 150 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 600 | 300 | 50 | 10 | 140 |
|  |  |  | 20 | 145 |
| 30 | 150 |
| 40 | 155 |
| 50 | 160 |
| 60 | 165 |
| 80 | 10 | 150 |
|  | 20 | 155 |
| 30 | 160 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 100 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 150 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 600 | 350 | 50 | 10 | 150 |
|  |  |  | 20 | 155 |
| 30 | 160 |
| 40 | 165 |
| 50 | 170 |
| 60 | 175 |
| 80 | 10 | 160 |
|  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |
| 50 | 180 |
| 60 | 185 |
| 100 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 150 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 600 | 400 | 50 | 10 | 160 |
|  |  |  | 20 | 165 |
| 30 | 170 |
| 40 | 175 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 50 | 180 |
| 60 | 185 |
| 80 | 10 | 170 |
|  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 100 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 150 | 10 | 190 |
|  | 20 | 195 |
| 30 | 200 |
| 40 | 205 |
| 50 | 210 |
| 60 | 215 |
| 600 | 450 | 50 | 10 | 170 |
|  |  |  | 20 | 175 |
| 30 | 180 |
| 40 | 185 |
| 50 | 190 |
| 60 | 195 |
| 80 | 10 | 180 |
|  | 20 | 185 |
| 30 | 190 |
| 40 | 195 |
| 50 | 200 |
| 60 | 205 |
| 100 | 10 | 190 |
|  | 20 | 195 |
| 30 | 200 |
| 40 | 205 |
| 50 | 210 |
| 60 | 215 |
| 150 | 10 | 200 |
|  | 20 | 205 |
| 30 | 210 |
| 40 | 215 |
| 50 | 220 |

Продолжение таблицы 2. Значения целевой функции в зависимости от значений входных параметров задачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 60 | 225 |

Рисунок 30 – Оптимальное решение целевой функции при Si = 100, Vi = 200

Рисунок 31 – Оптимальное решение целевой функции при Si = 100, Vi = 250

****

Рисунок 32 – Оптимальное решение целевой функции при Si = 100, Vi = 300

****

Рисунок 33 – Оптимальное решение целевой функции при Si = 100, Vi = 350

****

Рисунок 34 – Оптимальное решение целевой функции при Si = 100, Vi = 400

Рисунок 35 – Оптимальное решение целевой функции при Si = 100, Vi = 450

****

Рисунок 36 – Оптимальное решение целевой функции при Si = 200, Vi = 200

****

Рисунок 37 – Оптимальное решение целевой функции при Si = 200, Vi = 250

****

Рисунок 38 – Оптимальное решение целевой функции при Si = 200, Vi = 300

****

Рисунок 39 – Оптимальное решение целевой функции при Si = 200, Vi = 350

****

Рисунок 40 – Оптимальное решение целевой функции при Si = 200, Vi = 400

****

Рисунок 41 – Оптимальное решение целевой функции при Si = 200, Vi = 450

****

Рисунок 42 – Оптимальное решение целевой функции при Si = 300, Vi = 200

****

Рисунок 43 – Оптимальное решение целевой функции при Si = 300, Vi = 250

****

Рисунок 44 – Оптимальное решение целевой функции при Si = 300, Vi = 300

****

Рисунок 45 – Оптимальное решение целевой функции при Si = 300, Vi = 350

****

Рисунок 46 – Оптимальное решение целевой функции при Si = 300, Vi = 400

****

Рисунок 47 – Оптимальное решение целевой функции при Si = 300, Vi = 450

****

Рисунок 48 – Оптимальное решение целевой функции при Si = 400, Vi = 200

****

Рисунок 49 – Оптимальное решение целевой функции при Si = 400, Vi = 250

****

Рисунок 50 – Оптимальное решение целевой функции при Si = 400, Vi = 300

****

Рисунок 51 – Оптимальное решение целевой функции при Si = 400, Vi = 350

****

Рисунок 52 – Оптимальное решение целевой функции при Si = 400, Vi = 400

****

Рисунок 53 – Оптимальное решение целевой функции при Si = 400, Vi = 450

****

Рисунок 54 – Оптимальное решение целевой функции при Si = 500, Vi = 200

****

Рисунок 55 – Оптимальное решение целевой функции при Si = 500, Vi = 250

****

Рисунок 56 – Оптимальное решение целевой функции при Si = 500, Vi = 300

****

Рисунок 57 – Оптимальное решение целевой функции при Si = 500, Vi = 350

****

Рисунок 58 – Оптимальное решение целевой функции при Si = 500, Vi = 400

****

Рисунок 59 – Оптимальное решение целевой функции при Si = 500, Vi = 450

****

Рисунок 60 – Оптимальное решение целевой функции при Si = 600, Vi = 350

****

Рисунок 61 – Оптимальное решение целевой функции при Si = 600, Vi = 400

****

Рисунок 62 – Оптимальное решение целевой функции при Si = 600, Vi = 450

**Выводы по разделу 4**

В результате анализа эффективности применения моделей, направленных на оптимизацию распределенного хранения и обработки данных, можно сделать следующие выводы.

Первая математическая модель, основанная на смешанном целочисленном линейном программировании, была разработана для минимизации суммарной стоимости выполнения операций и учета ограничений на директивные сроки обработки данных. Анализ этой модели позволил получить оптимальные решения целевой функции благодаря определенным переменным. Эта модель может быть полезна в ситуациях, где важно учесть стоимостные ограничения и соблюсти требования по времени выполнения операций. Она позволяет достичь баланса между стоимостью и временем, обеспечивая оптимальное распределение ресурсов.

Вторая математическая модель также основана на смешанном целочисленном линейном программировании и направлена на минимизацию времени на реализацию операций при ограничении на стоимости их выполнения. Анализ этой модели позволяет оптимизировать процесс обработки данных, учитывая ограничения на стоимость реализации операций. Она может быть полезна в ситуациях, где важно сократить время обработки данных и при этом соблюсти бюджетные ограничения.

# **ЗАКЛЮЧЕНИЕ**

В ходе проведенного исследования в области планирования виртуальных машин в распределенных системах было подробно рассмотрено и обоснованы модели. Также была разработана модель, основанная на оптимизации распределенного хранения и распределенной обработки данных.

Разработанная модель оптимизации распределенного хранения и обработки данных является эффективным инструментом для повышения производительности распределенных систем. Она позволяет оптимально распределять ресурсы и задачи между виртуальными машинами, учитывая их требования и характеристики.

Основной целью работы было повышение эффективности планирования виртуальных машин в распределенных системах. Исследование подтвердило актуальность этой задачи и показало, что разработанная модель способствует улучшению общей производительности распределенных систем. В современных распределенных системах с большим количеством виртуальных машин и разнообразными рабочими нагрузками, правильное планирование становится необходимостью для эффективного использования ресурсов и достижения требуемых уровней производительности.

Оптимизация планирования виртуальных машин влияет на различные аспекты распределенных систем, включая загрузку ресурсов, производительность и надежность. Правильное распределение задач и ресурсов позволяет избежать перегрузки некоторых узлов и обеспечить балансировку нагрузки, что в конечном итоге приводит к повышению производительности и снижению вероятности отказов.

Без правильного планирования, распределенные системы могут столкнуться с рядом проблем, включая недостаточную загрузку ресурсов, низкую производительность и проблемы с надежностью.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Ghemawat, S., Gobioff, H., & Leung, S. T. (2003). The Google File System. ACM SIGOPS Operating Systems Review, 37(5), 29–43.

2. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1), 107–113.

3. Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M. … Stoica, I. (2012). Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing. Proceedings of the ninth USENIX Conference on Networked Systems Design and Implementation, 2–2.

4. Lakshman, A., & Malik, P. (2010). Cassandra: A Decentralized Structured Storage System. ACM SIGOPS Operating Systems Review, 44(2), 35–40.

5. Zaharia, M., Xin, R. S., Wendell, P., Das, T., Armbrust, M., Dave, A., Stoica, I. (2016). Apache Spark: A Unified Engine for Big Data Processing. Communications of the ACM, 59(11), 56–65.

6. Kairam, S., Diao, Y., Ranganathan, P., Malkowski, S., & Das, S. (2017). A comparative study of big data processing frameworks. Journal of Big Data, 4(1), 1–28.

7. Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The Hadoop Distributed File System. Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST), 1–10.

8. Распределительная система для хранения и обработки данных. URL: [https://www.atlassian.com/ru/microservices/microservices architecture/distributed-architecture](https://www.atlassian.com/ru/microservices/microservices%20architecture/distributed-architecture)

9. Шевченко С.В. Оптимизация архитектуры виртуальных вычислительных машин в распределенных системах обработки данных. URL: <https://cyberleninka.ru/article/n/optimizatsiya-arhitektury-virtualnyh-vychislitelnyh-mashin-v-raspredelennyh-sistemah-obrabotki-dannyh/viewer>

10. Селезнев С.В. Технологии виртуализации. URL: <https://moluch.ru/archive/55/7488/>

11. Usmani, Z.; Singh, S. A survey of virtual machine placement techniques in a cloud data center. Procedia Comput. Sci. 2016, 78, 491–498.

12. Dong, J.; Jin, X.; Wang, H.; Li, Y.; Zhang, P.; Cheng, S. Energy-saving virtual machine placement in cloud data centers. In Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2013), Delft, The Netherlands, 13–16 May 2013; pp. 618–624.

13. Silva Filho, M.C.; Monteiro, C.C.; Inácio, P.R.M.; Freire, M.M. Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. J. Parallel Distrib. Comput. 2018, 111, 222–250.

14. Greenberg, A.; Hamilton, J.R.; Jain, N.; Kandula, S.; Kim, C.; Lahiri, P.; Sengupta, S. VL2: A scalable and flexible data center network. ACM SIGCOMM Comput. Commun. Rev. 2009, 39, 51–62.

15. Stefanello, F.; Buriol, L.S.; Aggarwal, V.; Resende, M.G. A new linear model for placement of virtual machines across geo-separated data centers. Simpsio Bras. Pesqui. Oper. 2005, 47, 1–11.

16. Song, W.; Xiao, Z.; Chen, Q.; Luo, H. Adaptive resource provisioning for the cloud using online bin packing. IEEE Trans. Comput. 2014, 63, 2647–2660.

17. Beloglazov, A.; Buyya, R. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality-of-service constraints. IEEE Trans. Parallel Distrib. Syst. 2013, 24, 1366–1379.

18. Ghobaei-Arani, M.; Rahmanian, A.A.; Shamsi, M.; Rasouli-Kenari, A. A learning-based approach for virtual machine placement in cloud data centers. Int. J. Commun. Syst. 2018, 31.

19. Bobroff, N.; Kochut, A.; Beaty, K. Dynamic placement of virtual machines for managing SLA violations. In Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management, Munich, Germany, 21–25 May 2007; pp. 119–128.

20. Zheng, Q.; Li, R.; Li, X.; Shah, N.; Zhang, J.; Tian, F.; Chao, K.M.; Li, J. Virtual machine consolidated placement based on multi-objective biogeography-based optimization. Future Gener. Comput. Syst. 2016, 54, 95–122.

21. Van, H.N.; Tran, F.D.; Menaud, J.-M. Performance and power management for cloud infrastructures. In Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD 2010), Miami, FL, USA, 5–10 July 2010; pp. 329–336.

22. Liao, X.; Jin, H.; Liu, H. Towards a green cluster through dynamic remapping of virtual machines. Future Gener. Comput. Syst. 2012, 28, 469–477.

23. Guo, C.; Lu, G.; Wang, H.J.; Yang, S.; Kong, C.; Sun, P.; Wu, W.; Zhang, Y. SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees. In Proceedings of the 6th International Conference (Co-NEXT ’10), Philadelphia, PA, USA, 30 November–3 December 2010. Article No. 15.

24. Ballani, H.; Costa, P.; Karagiannis, T.; Rowstron, A. Towards predictable datacenter networks. ACM SIGCOMM Comput. Commun. Rev. 2011, 41, 242–253.

25. Xie, D.; Hu, Y.C. The only constant is change: Incorporating time-varying network reservations in data centers. ACM SIGCOMM Comput. Commun. Rev. 2012, 42, 199–210.

26. Stefanello, F.; Aggarwal, V.; Buriol, L.S.; Gonçalves, J.F.; Resende, M.G. A biased random-key genetic algorithm for placement of virtual machines across geo-separated data centers. In Proceedings of the 2015 Conference on Genetic and Evolutionary Computation (GECCO’15), Madrid, Spain, 11–15 July 2015; pp. 919–926.

# **ПРИЛОЖЕНИЕ А. КОД ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

from ortools.linear\_solver import pywraplp  
  
# Создаем линейный решатель с помощью GLOP (Google Linear Optimization Package).  
solver = pywraplp.Solver.CreateSolver('GLOP')  
  
# Определяем переменные  
L = 5  
I = 5  
M = 5  
x\_s = {}  
x\_c = {}  
x\_i = {}  
for l in range(L):  
 for i in range(I):  
 x\_s[l, i] = solver.NumVar(0, 1, f'x\_s[{l},{i}]')  
 x\_c[l, i] = solver.NumVar(0, 1, f'x\_c[{l},{i}]')  
 for m in range(M):  
 x\_i[l, m, i] = solver.NumVar(0, 1, f'x\_i[{l},{m},{i}]')  
  
# Определяем коэффициенты  
q\_l = [5] \* L  
v\_i = 300  
c\_m = [5] \* M  
s\_i = 300  
w\_lm = [5] \* L \* M  
b\_lm = [[1] \* M for \_ in range(L)]  
p\_im = [[1] \* M for \_ in range(I)]  
V\_l = [5] \* L  
P\_m = 150,200,250,100,300  
d\_i = 50  
  
# Определяем целевую функцию  
objective = solver.Objective()  
for l in range(L):  
 for i in range(I):  
 objective.SetCoefficient(x\_s[l, i], q\_l[l] \* v\_i[i])  
for m in range(M):  
 for i in range(I):  
 objective.SetCoefficient(x\_c[m, i], c\_m[m] \* s\_i[i] \* p\_im[i][m])  
 for i in range(I):  
 for m in range(M):  
 for l in range(L):  
 objective.SetCoefficient(x\_i[l, m, i],w\_lm[l \* M + m] \* v\_i[i] \* b\_lm[l][m])  
objective.SetMinimization()  
  
# Определяем ограничения  
for l in range(L):  
 for i in range(I):  
 for m in range(M):  
 ct1 = solver.Constraint(-solver.infinity(), 0)  
 ct1.SetCoefficient(x\_i[l, m, i], 1)  
 ct1.SetCoefficient(x\_s[l, i], -1)  
  
 ct2 = solver.Constraint(-solver.infinity(), 0)  
 ct2.SetCoefficient(x\_i[l, m, i], 1)  
 ct2.SetCoefficient(x\_c[m, i], -1)  
  
 ct3 = solver.Constraint(-solver.infinity(), 0)  
 ct3.SetCoefficient(x\_i[l, m, i], -1)  
 ct3.SetCoefficient(x\_s[l, i], 1)  
 ct3.SetCoefficient(x\_c[m, i], 1)  
 ct3.SetUb(1)  
  
 for i in range(I):  
 ct4 = solver.Constraint(1, 1)  
 for l in range(L):  
 ct4.SetCoefficient(x\_s[l, i], 1)  
  
 for i in range(I):  
 ct5 = solver.Constraint(1, 1)  
 for m in range(M):  
 ct5.SetCoefficient(x\_c[l, i], 1)  
  
 for l in range(L):  
 ct6 = solver.Constraint(-solver.infinity(), V\_l[l])  
 for i in range(I):  
 ct6.SetCoefficient(x\_s[l, i], v\_i[i])  
  
 for m in range(M):  
 ct7 = solver.Constraint(-solver.infinity(), P\_m[m])  
 for i in range(I):  
 ct7.SetCoefficient(x\_c[m, i], p\_im[i][m])  
  
 for i in range(I):  
 ct8 = solver.Constraint(-solver.infinity(), d\_i[i])  
 for l in range(L):  
 for m in range(M):  
 ct8.SetCoefficient(x\_i[l, m, i], v\_i[i] \* b\_lm[l][m])  
 for m in range(M):  
 ct8.SetCoefficient(x\_c[m, i], s\_i[i] \* p\_im[i][m])  
  
# Решаем модель и получаем решение  
status = solver.Solve()  
if status == pywraplp.Solver.OPTIMAL:  
 print('Оптимальное значение:', solver.Objective().Value())  
else:  
 print('Данная целевая функция с ограничениями не имеет оптимального решения')  
  
# Выводим решения для x\_il^s

print('Решение для x\_il^s:')

for l in range(L):

x\_il\_s = []

for i in range(I):

x\_il\_s.append(int(x\_s[l, i].solution\_value()))

print(x\_il\_s)

# Выводим решения для x\_mi^c

print('Решение для x\_mi^c:')

for m in range(M):

x\_mi\_c = []

for i in range(I):

x\_mi\_c.append(int(x\_c[m, i].solution\_value()))

print(x\_mi\_c)

# Выводим решения для p\_im^c

print('Решение для p\_im^c:')

for i in range(I):

p\_im\_c = []

for m in range(M):

p\_im\_c.append(int(p\_im[i][m] \* x\_c[m, i].solution\_value()))

print(p\_im\_c)