

На правах рукописи



Сахно Мария Юрьевна

**МАТЕМАТИЧЕСКИЕ МОДЕЛИ И СПИСОЧНЫЕ АЛГОРИТМЫ  
ДЛЯ ПОСТРОЕНИЯ РАСПИСАНИЙ  
В МНОГОПРОЦЕССОРНЫХ СИСТЕМАХ  
С РЕСУРСНЫМИ ОГРАНИЧЕНИЯМИ**

1.2.2. Математическое моделирование, численные методы  
и комплексы программ

Автореферат  
диссертации на соискание учёной степени  
кандидата технических наук

Омск — 2025

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институт математики им. С.Л. Соболева Сибирского отделения Российской академии наук.

Научный руководитель: **Захарова Юлия Викторовна**, кандидат физико-математических наук, старший научный сотрудник лаборатории дискретной оптимизации Омского филиала Федерального государственного бюджетного учреждения науки Института математики им. С.Л. Соболева Сибирского отделения Российской академии наук, г. Омск

Официальные оппоненты: **Казаковцев Лев Александрович**, доктор технических наук, профессор, профессор кафедры системного анализа и исследования операций Федерального государственного бюджетного образовательного учреждения высшего образования «Сибирский государственный университет науки и технологий имени академика М.Ф. Решетнева», г. Красноярск

**Бахтин Владимир Александрович**, кандидат физико-математических наук, ведущий научный сотрудник отдела №17 Федерального государственного учреждения «Федеральный исследовательский центр Институт прикладной математики им. М.В. Келдыша Российской академии наук», г. Москва

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук, г. Новосибирск

Защита состоится 24.12.2025 в 14:00 на заседании диссертационного совета 24.2.350.05, созданного на базе Федерального государственного автономного образовательного учреждения высшего образования «Омский государственный технический университет» по адресу: 644050, г. Омск, пр. Мира, д. 11, ауд. П-202.

С диссертацией можно ознакомиться в библиотеке Омского государственного технического университета и на официальном сайте <http://www.omgtu.ru>.

Автореферат разослан «\_\_» \_\_\_\_\_ 2025 года.

Ученый секретарь  
диссертационного совета 24.2.350.05,  
доктор технических наук, доцент



Варепо Л.Г.

## Общая характеристика работы

**Актуальность темы.** В данной работе исследуются математические модели, связанные с задачами составления расписаний, возникающими в многопроцессорных компьютерных системах, например, при разработке программы для выполнения на многоядерном процессоре. Многопроцессорные системы характеризуются такими свойствами, как наличие общего ресурса и возможность распараллеливания вычислений. Математические модели для составления расписаний выполнения подпрограмм (работ) на процессорах или ядрах процессора должны учитывать оба эти свойства. Задачи оптимизации, возникающие в рамках этих моделей, актуальны для производителей процессоров и компаний, разрабатывающих многопоточное программное обеспечение, т.к. их решение может повысить скорость работы программного обеспечения.

Работы могут влиять друг на друга при совместном выполнении из-за наличия общего ресурса. Например, скорость выполнения работы может меняться в зависимости от загрузки других ядер процессора в случае, когда разным работам необходимо передавать разный объем данных по шине данных. Может возникнуть конкуренция за шину данных и выполнение каждой из работ в этом случае может занять больше времени, чем в случае однопоточного выполнения. Необходимо составить расписание выполнения работ на ядрах процессора, учитывая их взаимное влияние друг на друга. Длительности работ также зависят от скорости, с которыми они выполняются, что влияет на общее потребление такого ресурса, как энергия: чем с большей скоростью выполняется работа, тем больше энергии на неё затрачивается, но тем меньше её длительность. В этом случае необходимо распределить энергию между работами таким образом, чтобы минимизировать её расход или уложиться в заданные границы.

Характерным свойством многопроцессорных систем является распараллеливание. Это означает, что каждая работа может выполняться на двух и более процессорах одновременно (также можно рассматривать ядра). Существует несколько основных типов, определяющих степень распараллеливания работ: задано необходимое количество процессоров (англ. rigid) или задана верхняя граница на число используемых процессоров, при этом фактическое количество определяется перед началом выполнения работы (англ. moldable) или может изменяться в процессе выполнения (англ. malleable). Это также важно учитывать при составлении расписания выполнения работ.

Многопроцессорные компьютерные системы часто реализуются с использованием NUMA-архитектуры (англ. Non-Uniform Memory Access). В таких системах процессорные ядра и модули оперативной памяти объединены в NUMA-узлы, внутри которых доступ к памяти осуществляется быстрее, чем к памяти других узлов. При этом задачи и процессы могут быть размещены как в пределах одного NUMA-узла, так и с использованием ресурсов нескольких узлов одновременно. Эта особенность также важна при размещении виртуальных

машин на серверах, где ресурсы (ядра процессора и объём памяти) распределены между NUMA-узлами. Для эффективного использования аппаратных ресурсов и минимизации задержек критично учитывать NUMA-архитектуру.

**Степень разработанности темы.** В литературе существует ряд подходов к планированию назначения работ на ядра процессора с учетом переменной длительности их выполнения. Как правило, такие задачи решаются с помощью быстрых эвристических алгоритмов, которые работают в онлайн-режиме, т.е. работы поступают последовательно и в каждый момент времени рассматривается только ограниченное количество работ. Эвристические алгоритмы для планирования работ, предложенные А. Merkel, С. Журавлевым и другими, используют стратегию, которая старается размещать работы на ядра процессора комплементарным образом, чтобы работы с наиболее различными потребностями в использовании ресурсов выполнялись одновременно (например, такими ресурсами могут быть пропускная способность шины данных и кэш на разных уровнях).

Задача размещения виртуальных машин по серверам представляет собой обобщение темпоральной задачи упаковки в контейнеры, в которой каждый предмет занимает ресурсы в течение заданного временного интервала [N. Aydın, 2020]. Для её решения используются как точные методы, например, основанные на ветвлении [M. Dell’Amico, 2020], так и приближённые подходы: жадные эвристические алгоритмы [M. Delorme, 2016], метод генерации столбцов [A. Ratushnyi, 2021], генетический алгоритм [M. Sakhno, 2023] и другие.

Для задач, где ресурсом выступает энергия, известно много теоретических исследований. К. Pruhs исследовал задачу минимизации среднего времени выполнения работ в однопроцессорной системе при фиксированном количестве энергии и с заданным временем поступления каждой работы. Он предложил алгоритм с полиномиальным временем для выполнения работ идентичного объема. D. Vunde применил этот подход к задаче с несколькими процессорами и работами произвольного объема. F. Yao исследовал задачу с критерием минимизации потребляемой энергии с одним процессором и с возможностью прерывать выполнение работ и предложил точный алгоритм YDS, оригинальная вычислительная сложность которого  $O(n^3)$ , где  $n$  – это количество работ. Если же рассмотреть задачу с несколькими процессорами и распараллеливаемыми работами, то такая задача становится NP-трудной даже если прерывания допустимы [А.В. Кононов, 2020]. А.В. Кононов также провел анализ вычислительной сложности и предложил двухэтапные конструктивные алгоритмы для построения расписаний для произвольного числа процессоров с учетом распараллеливаемых работ и наличием энергии.

Для задач составления расписаний, возникающих в многопроцессорных компьютерных системах, является актуальной разработка метаэвристик, среди которых есть класс эволюционных алгоритмов, хорошо зарекомендовавших себя при решении задач составления расписаний с ресурсными ограничениями.

Генетический алгоритм является эволюционным эвристическим алгоритмом, который имитирует процесс естественной эволюции [C.R. Reeves, 1997]. Чтобы поддерживать достаточный уровень разнообразия популяции применяют, например, механизм перезапуска алгоритма или более интенсивную мутацию [B. Doerr, 2024]. Для осуществления направленного поиска хорошо себя зарекомендовали оптимизированные операторы скрещивания, которые строят лучшего потомка, удовлетворяющего тем или иным свойствам [C. Aggarwal, 1997]. Значения параметров алгоритма выбираются на этапе препроцессинга или адаптируются в процессе эволюции [M. Drugan, 2019]. Адаптивное управление вызовом операторов использует обратную связь из истории поиска для определения направления дальнейшего поиска. Существуют самоадаптирующиеся (англ. self-adaptive) варианты настройки параметров [T. Back, 1998]. В данной схеме настраиваемые параметры включены в кодировку особей и также изменяются в процессе эволюции.

**Целью** данной работы является выявление свойств математических моделей и создание вычислительных методов и комплексов программ, ориентированных на оптимизацию составления расписаний в многопроцессорных компьютерных системах с учетом ресурсных ограничений, и повышение эффективности решения практических задач.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Анализ комбинаторных свойств рассматриваемых задач составления расписаний в рамках моделей частичного целочисленного программирования.
2. Исследование вычислительной сложности задач и разработка конструктивных алгоритмов списочного типа, учитывающих специфику задач и позволяющих быстро находить допустимые решения.
3. Разработка и анализ адаптивного эволюционного алгоритма для решения рассматриваемых задач составления расписаний с учетом ограничения на потребление общего ресурса и свойства распараллеливания.
4. Создание комплекса программ, ориентированных на решение поставленных задач. Проведение вычислительных экспериментов.

**Научная новизна:**

1. Получены новые свойства допустимых решений рассматриваемых задач на основе исследования оригинальных моделей частично целочисленного программирования, использующих концепцию точек событий с непрерывным представлением времени.
2. Доказана NP-трудность задачи составления расписаний с учетом пропускной способности шины данных и задачи составления расписаний при возможности распараллеливания операций с ограничением на расход энергии. Выявлены комбинаторные свойства задач, позволившие сократить трудоемкость предлагаемых алгоритмов для их решения.

3. Разработаны конструктивные алгоритмы списочного типа для решения рассматриваемых задач составления расписаний в компьютерных системах с учетом ресурсов и свойства распараллеливания. Предложенные алгоритмы имеют статистически значимое преимущество перед известными аналогами.

4. Разработан адаптивный эволюционный алгоритм с оптимизированными операторами для составления расписаний в компьютерных системах с учетом ограничения на потребление общего ресурса и свойства распараллеливания. Данный алгоритм также может использоваться и для других задач составления расписаний на перестановках.

#### **Теоретическая значимость:**

1. Доказана NP-трудность задачи составления расписаний на многоядерных процессорах с учетом пропускной способности шины данных.

2. Доказана NP-трудность задачи составления расписаний в многопроцессорных компьютерных системах с учетом расхода энергии и свойства распараллеливания работ.

3. Доказана сходимости предложенного адаптивного эволюционного алгоритма к оптимуму для решения рассматриваемых задач.

4. Выявлены комбинаторные и оптимизационные свойства математических моделей рассматриваемых задач.

**Практическая значимость** Разработанные конструктивные алгоритмы и адаптивный эволюционный алгоритм для рассматриваемых задач протестированы на сериях тестовых примеров, аналогичных возникающим на практике. Результаты экспериментов показали конкурентное преимущество по сравнению с алгоритмическими пакетами в составе известных решателей (Gurobi, CPLEX) и онлайн-планировщиком oneTBV. Результаты диссертации могут быть использованы для улучшения планировщиков работы приложений и производителей процессоров, а также для повышения эффективности планирования работ для серверов облачных ресурсов. Разработанные в диссертации конструктивные алгоритмы и выявленные свойства расписаний апробированы и внедрены в прикладную деятельность ООО «Техкомпания Хуавей» при анализе системы планирования работ.

**Объект исследования** – расписания выполнения работ на многоядерных процессорах.

**Предметом исследования** являются математические модели, методы и комплексы программ составления расписаний выполнения работ на многоядерных процессорах.

**Методология и методы исследования.** Обоснованность и достоверность научных результатов и выводов, содержащихся в данной работе, базируются на фундаментальных положениях целочисленного программирования, теории вероятностей и математической статистики, теории вычислительной сложности, методах математического моделирования, а также применении

современных компьютерных технологий и методологии экспериментальных исследований.

**Основные положения, выносимые на защиту:**

1. Выявлены сложностные и комбинаторные свойства конфигураций работ, позволившие разработать конструктивные алгоритмы списочного типа и адаптивный эволюционный алгоритм для составления расписаний выполнения работ с учетом ограничения на потребление общего ресурса и свойства распараллеливания.

2. Разработан эффективный конструктивный алгоритм построения приближенного решения с проблемно-ориентированной стратегией жадного типа назначения работ на ядра процессора, позволяющий учитывать пропускную способность шины данных при планировании выполнения работ в многопоточных системах.

3. Предложен эффективный метод локальных улучшений структурных компонент решений для повышения качества расписаний с возможностью распараллеливания работ и ограничением на расход энергии.

4. Создан комплекс программ, реализующий предложенные математические модели, конструктивные алгоритмы и метаэвристики для составления расписаний в компьютерных системах с учетом ограничения на потребление общего ресурса и свойства распараллеливания, а также модуль запуска планировщика работы приложений.

**Соответствие научной специальности.** Работа соответствует научной специальности 1.2.2 по п. 3 – Реализация эффективных численных методов и алгоритмов в виде комплексов проблемно-ориентированных программ для проведения вычислительного эксперимента; по п. 7 – Качественные или аналитические методы исследования математических моделей; по п. 9 – Постановка и проведение численных экспериментов, статистический анализ их результатов, в том числе с применением современных компьютерных технологий.

**Достоверность** научных положений, выводов и практических рекомендаций, полученных в диссертации, подтверждается корректным обоснованием постановок задач, точной формулировкой критериев, достаточным количеством численных экспериментов с последующим статистическим анализом, математическими доказательствами теоретических утверждений. Методика проведения численных экспериментов подробно описана, что позволяет воспроизвести полученные результаты.

**Апробация работы.** Основные результаты диссертации докладывались на следующих конференциях и семинарах:

– международная конференция «Mathematical Optimization Theory and Operations Research» (2021, 2023, 2024, 2025)

– международная научная конференция «Математическое и компьютерное моделирование» (2024, 2025)

- международная конференция «Numerical Computations: Theory and Algorithms» (NUMTA, 2023)
- международная конференция и молодежная школа «Математическое моделирование и суперкомпьютерные технологии» (2024)
- международная конференция «Optimization and Applications» (OPTIMA, 2020, 2022)
- региональная конференция магистрантов, аспирантов и молодых ученых по физике, математике и химии «ФМХ ОмГУ» (2022)
- семинар ОФ ИМ СО РАН «Модели и алгоритмы для задач составления расписаний» (2022, 2023, 2024, 2025)
- семинар ОФ ИМ СО РАН «Математическое моделирование и дискретная оптимизация» (2020, 2022, 2023, 2024, 2025)

**Личный вклад.** Решение задач диссертации, разработанные алгоритмы и их программная реализация, экспериментальные и теоретические результаты, представленные в диссертации и выносимые на защиту, принадлежат лично автору.

**Публикации.** Основные результаты диссертации опубликованы в 14 научных работах, две из них изданы в журналах из списка ВАК, одна – в периодических научных журналах, индексируемых Web of Science и Scopus, три – в трудах международных конференций, индексируемых в библиографических зарубежных базах данных публикаций (2020, 2024, 2025). Зарегистрирован один акт о внедрении, получено три свидетельства о государственной регистрации программ для ЭВМ. Конфликт интересов с соавторами отсутствует.

**Объем и структура работы.** Диссертация состоит из введения, 4 глав, заключения и 3 приложений. Полный объем диссертации составляет 145 страниц, включая 34 рисунка и 38 таблиц. Список литературы содержит 128 наименований.

## Содержание работы

Во **введении** обоснована актуальность исследуемой проблемы, сформулированы цели и задачи диссертационной работы, показана новизна работы и представлены основные положения, выносимые на защиту, описана структура диссертации.

В **первой главе** проводится анализ задач составления расписаний выполнения работ в многоядерных системах с учетом их взаимного замедления и особенностей NUMA-архитектуры.

Имеется множество работ  $\mathcal{J} = \{1, \dots, n\}$  и  $m$  ядер процессора. Прерывание выполнения любой работы запрещено. Работы не меняют ядро в процессе выполнения. На одном ядре не может выполняться более одной работы. На множестве работ также задан частичный порядок.

Для каждой работы  $j \in \mathcal{J}$ , известно количество единиц времени  $\tilde{p}_j$ , необходимое ей для полного выполнения в идеальных условиях (т.е. при условии, что вместе с ней не выполняются другие работы).

Известно, что работы при одновременном выполнении могут замедлять друг друга за счет совместного использования ресурсов. Такие задачи возникают, в частности, при разработке планировщиков параллельных приложений, таких как oneTBB. В этом случае работы соответствуют процедурам (программным модулям), которые необходимо назначить ядрам процессора на выполнение. К ресурсам, конкуренция за которые приводит к замедлению, могут относиться кэш-память различных уровней и пропускная способность системной шины. Задача рассматривается в двух вариантах: когда для каждого подмножества работ, которые могут выполняться вместе (с учетом частичного порядка), задано, как они замедляют друг друга, либо в численном виде, либо в виде правила для вычисления замедлений на основе потребления ими общего ресурса.

### **Задача построения расписания для многоядерного процессора с учетом взаимного влияния работ**

Для постановки задачи в первом случае необходимо ввести определения *конфигурации* и *скорости выполнения работы*. *Конфигурация* – это набор работ, выполняющихся одновременно на разных ядрах с учетом частичного порядка на множестве работ и ограничений на количество ядер. Множество всех конфигураций обозначается как  $\mathcal{C}$ . *Скоростью выполнения работы  $j \in \mathcal{J}$  в конфигурации  $c \in \mathcal{C}$*  называется отношение времени выполнения работы  $j$  в идеальных условиях ко времени полного выполнения работы  $j$ , если бы  $j$  всё это время выполнялась в конфигурации  $c$ . Скорость работы зависит от конфигурации, в которой она выполняется. На протяжении каждой конфигурации скорость выполнения всех работ считается постоянной. Итак, для каждой конфигурации  $c \in \mathcal{C}$  известно, из каких работ она состоит. Для каждой работы  $j$  в конфигурации  $c$  известна скорость её выполнения  $v_{jc}$ .

Необходимо составить такое расписание выполнения работ на ядрах процессора, что общее время завершения работ  $C_{\max}$  минимально.

В главе доказаны следующие теоремы:

**Теорема 1.** *Задача построения расписания для многоядерного процессора с учетом взаимного влияния работ является NP-трудной при числе ядер  $t = 2$ .*

**Теорема 2.** *Задача построения расписания для многоядерного процессора с учетом взаимного влияния работ является NP-трудной в сильном смысле при числе ядер  $t \geq 3$ , ограниченном константой.*

Для указанной задачи построена модель частично целочисленного линейного программирования (ЧЦЛП), использующая концепцию точек событий. В

настоящей главе точке событий соответствует реализация конфигурации работ. Основной особенностью предложенной модели является формирование последовательности выполнения выбранных конфигураций на основе предварительных вычислений скоростей работ в этих конфигурациях и учетом частичного порядка.

Доказано следующее утверждение:

**Утверждение 1.** *Для нахождения оптимального расписания выполнения  $n$  работ необходимо не более, чем  $2n + 1$  точек событий.*

### **Задача построения расписания для многоядерного процессора с учетом потребления шины данных**

Поскольку количество конфигураций в задаче построения расписания для многоядерного процессора с учетом взаимного влияния работ может быть довольно большим (до  $\sum_{i=0}^m C_n^i$  в зависимости от частичного порядка), постановку задачи можно упростить, введя предположение о том, что скорости выполнения работ рассчитываются исходя из фактического потребления ими шины данных. В этой постановке для каждой работы  $j \in \mathcal{J}$  необходимо знать процент потребления шины данных  $\tilde{b}_j$  при выполнении в идеальных условиях и некоторое правило, согласно которому можно рассчитать замедления работ при совместном выполнении.

В главе доказано, что эта задача также является NP-трудной при числе ядер  $m = 2$  и NP-трудной в сильном смысле при числе ядер  $m \geq 3$ , ограниченном константой.

Предложен жадный алгоритм её решения (алгоритм 1), а в качестве правила по расчету замедления работ  $j \in \mathcal{J}$  в конфигурации  $c \in C$  при известных значениях потребления ими шины данных  $\tilde{b}_j$  представлен алгоритм, который поэтапно распределяет пропускную способность шины данных между работами из одной конфигурации так, чтобы каждая работа получала не больше своей потребности, а оставшийся ресурс равномерно перераспределялся между оставшимися.

---

### **Алгоритм 1** Жадный алгоритм для задачи построения расписания для многоядерного процессора с учетом потребления шины данных

---

- 1: **пока** присутствуют незавершенные работы **выполнять**
  - 2: Сформировать конфигурацию, максимально загружая ядра процессора (с учётом частичного порядка). Работы выбираются таким образом, чтобы их использование шины данных было наиболее близко к оставшейся свободной шине данных.
  - 3: Распределить пропускную способность шины данных между работами из выбранной конфигурации и вычислить скорости выполнения работ.
  - 4: Определить длительность конфигурации до завершения первой работы, оставшиеся работы перенести в следующую конфигурацию.
  - 5: **конец пока**
-

Жадный алгоритм строит допустимое расписание и имеет временную сложность  $O(n^2)$ . В качестве конкурента жадного алгоритма можно рассматривать списочный алгоритм, который в отличие от жадного определяет порядок назначения работ в соответствии с переданной перестановкой с учетом ограничений частичного порядка.

### Задача назначения работ на машины с NUMA-архитектурой с учетом возобновимых ресурсов

Также в современных компьютерных системах актуальна задача размещения виртуальных машин по серверам, где работы также конкурируют за потребление ресурсов. Задача заключается в размещении виртуальных машин (работ) на серверах (машинах), при этом каждая виртуальная машина требует определённое количество ресурсов (количество ядер и памяти) и работает в заданном интервале времени, а серверы имеют NUMA-архитектуру, то есть состоят из NUMA-узлов с ограниченными ресурсами. Виртуальные машины с небольшими потребностями в ресурсах размещаются на одном узле, а остальные равномерно размещаются на нескольких NUMA-узлах. Необходимо минимизировать максимальное количество одновременно активных серверов. Таким образом, заданы  $\mathcal{J}^s$ ,  $\mathcal{J}^l$  – множества маленьких и больших работ,  $\mathcal{M}$  – множество машин,  $\mathcal{N}$  – множество NUMA-узлов машины,  $\mathcal{R}$  – множество ресурсов и  $\mathcal{T}$  – множество событий (моменты начала или окончания работ). Также известны  $c_\rho$  – количество ресурса типа  $\rho$ , доступного на одном NUMA-узле,  $\alpha_{j\rho}$  – потребность работы  $j$  в ресурсе  $\rho$ ,  $r_j$ ,  $d_j$  – моменты начала и окончания работы  $j$ .

Рассматриваемая задача является NP-трудной в сильном смысле, для неё известна модель целочисленного линейного программирования [Ратушный А., 2021]. В диссертации для этой задачи предложен алгоритм списочного типа. Алгоритм последовательно обрабатывает работы в заданном порядке и размещает их на подходящей машине с достаточным объёмом ресурсов. Порядок работ может определяться, например, по времени запуска или по весу, зависящему от ресурсов (ядра и память). Для выбора подходящей машины применяются стратегии First Fit, Random Fit и др. Адаптация к NUMA-архитектуре выполняется двумя способами: либо при размещении работы сразу выбирается подходящий NUMA-узел (или несколько – для больших работ), либо сначала работы размещаются по машинам, а затем отдельным алгоритмом – по NUMA-узлам. Стратегии выбора узла включают выбор наименее/наиболее загруженного, случайный выбор и равномерное распределение нагрузки.

Для проведения вычислительного эксперимента на первых двух задачах были построены серии тестовых примеров для разного количества работ, разного количества ядер и с разным типом частичного порядка: тривиальный частичный порядок **No-PO** (т.е. без связей между работами), случайный частичный порядок **R-PO**, бинарное дерево **B-PO** и один-ко-многим-к-одному **O-PO** (сначала одна работа, после нее много работ, после них вновь одна). В качестве работ выступают процедуры из библиотеки oneMKL (Intel oneAPI

Math Kernel Library). Здесь и далее для статистически значимых сравнений используется тест Вилкоксона.

Результаты жадного алгоритма сравниваются с нижней оценкой на целевую функцию, рассчитанной как  $\max\left(\max Path, \frac{\sum_{j \in \mathcal{J}} \tilde{p}_j}{m}\right)$ , где  $\max Path$  – самая длинная по суммарной длительности цепочка работ, связанных частичным порядком. Результаты жадного алгоритма зависят от типа частичного порядка. Для случайного частичного порядка, где наибольшее количество связей между работами, жадный алгоритм находит решения, совпадающие с нижней оценкой. В случае тривиального частичного порядка для 100 работ и 24 ядер жадный алгоритм всегда загружает все ядра, тем самым работы в значительной степени замедляют друг друга и среднее относительное отклонение его решений от нижней границы составляет 65,18%. Для двенадцати совместно выполняемых работ их взаимное влияние существенно снижается и среднее относительное отклонение составляет 13,74%.

Модификации предложенного выше жадного алгоритма могут быть основаны на том, что работы добавляются в расписание в соответствии с заданным порядком. В случае 10 работ такие модификации не привели к статистически значимым улучшениям, но на размерности задач в 50 и 100 работ лучшие результаты показал алгоритм списочного типа  $LT_p$ , который упорядочивает работы по невозрастанию  $\tilde{p}_j$  (количество единиц времени, необходимо работе  $j$  для выполнения в идеальных условиях). В этом случае жадный алгоритм не определяет последовательность выполнения работ, а берет их из переданной последовательности. Среднее относительное отклонение решений алгоритма  $LT_p$  от рекорда по каждому примеру составляет 0,99%.

Результаты данной главы опубликованы в [1; 8; 3] и [7].

Во **второй главе** рассматривается задача, в которой работы с жестким ограничением степени распараллеливания (англ. rigid) должны быть запланированы на  $m$  быстродействующих масштабируемых процессорах. Множество работ обозначается  $\mathcal{J} = \{1, \dots, n\}$ . Объем работы  $V_j$  и количество  $size_j$  требуемых процессоров указаны для каждой работы  $j \in \mathcal{J}$ . Значение  $size_j$  для работы  $j \in \mathcal{J}$  указывает требуемое количество процессоров для выполнения  $j$ . Общий объем  $V_j$  работы  $j \in \mathcal{J}$  равномерно распределяется между назначенными ей процессорами, т.е. если работа  $j$  использует  $size_j$  процессоров, то объемы обработки одинаковы для всех этих процессоров (обозначаются  $W_j := \frac{V_j}{size_j}$ ), и эти процессоры выполняют работу одновременно с одинаковой скоростью. Работы выполняются без прерываний и не меняют набор процессоров в процессе выполнения.

Также учитывается расход энергии, который определяется нелинейной функцией. Если процессор работает со скоростью  $s$ , то мгновенное потребление энергии в единицу времени (*мощность*) равно  $s^\alpha$ , где  $\alpha > 1$  является константой (например,  $\alpha = 1,11$  для Intel PXA270,  $\alpha = 1,66$  для механизма разгрузки

TSP,  $\alpha = 3$  для CMOS-устройств). Предполагается, что доступен непрерывный спектр скоростей процессоров.

Задача рассматривается в двух вариантах:

- когда минимизируется метрика качества расписания с бюджетными ограничениями на расход энергии;
- когда минимизируется расход энергии при наличии моментов поступления и директивных сроков работ.

Ограничение на потребление энергии возникает в случае, когда мощность батареи ограничена, то есть такая задача актуальна в приложениях, связанных с вычислительными устройствами, срок службы которых зависит от ограниченной эффективности батареи (например, многоядерные ноутбуки). Более того, в реальной практике возникают двухкритериальные задачи минимизации потребления энергии и некоторой метрики качества расписания. Возможный подход к решению – ограничить одну из целевых функций (например, потребление энергии) и оптимизировать другую. Задача составления расписания на двух процессорах актуальна, например, для компьютерных систем с конфигурацией из двух NUMA-узлов или устройств с двумя ядрами.

### **Задача составления расписания выполнения распараллеливаемых работ на процессорах с ограничением на расход энергии**

Необходимо составить расписание выполнения всех работ на процессорах такое, что суммарное время завершения всех работ  $\sum C_j$  при заданном энергетическом бюджете  $E$  и  $m = 2$  минимально.

### **Задача построения энергоэффективных расписаний выполнения работ на процессорах**

Необходимо составить расписание выполнения всех работ на одном процессоре такое, что суммарное потребление энергии минимально, при заданных для каждой работы моментов поступления (англ. release dates)  $r_j$  и директивных сроков (англ. deadlines)  $d_j$ .

Для обеих задач построены модели частично целочисленного выпуклого программирования, использующие концепцию точек событий. В настоящей главе точка событий – подмножество переменных модели, которое характеризует определенный набор работ с их временем начала и завершения и скоростью. В одной точке событий на каждом процессоре может выполняться не более одной работы. Множество всех точек событий обозначается как  $K = \{1, \dots, k_{\max}\}$ , где  $k_{\max} \leq n$  выбирается на основе предварительных оценок или предварительного эксперимента.

В главе доказана теорема:

**Теорема 3.** *Задача составления расписания выполнения распараллеливаемых работ на процессорах с ограничением на расход энергии NP-трудна.*

Для задачи составления расписания выполнения распараллеливаемых работ на процессорах с ограничением на расход энергии предложен алгоритм

списочного типа с локальными улучшениями (алгоритм 2), который дополняет результаты алгоритма  $LT_V^2$ , предложенного Кононовым А.В. и Захаровой Ю.В. в 2023 году, локальными улучшениями компонент решения.

В алгоритме 2 используется терминология блоков. Каждый блок содержит одну двухпроцессорную работу и те однопроцессорные работы, которые находятся между текущей и предыдущей двухпроцессорными. Блок считается нечетным, если он содержит нечетное количество однопроцессорных работ. Во второй главе диссертации получены новые свойства расписаний с учетом блочной структуры: получены достаточные условия, при которых перенос однопроцессорной работы из одного блока в другой приводит к улучшению качества расписания; доказано свойство, что упорядочение однопроцессорных работ по неубыванию объемов в пределах блока на каждом процессоре не приводит к ухудшению качества расписания.

---

**Алгоритм 2** Алгоритм списочного типа с локальными улучшениями.

---

- 1: Построить расписание алгоритмом  $LT_V^2$  и найти блоки в решении.
  - 2: Последовательно просматривать блоки в решении из Шага 1, применяя локальные улучшения между блоками.
  - 3: Применить локальные улучшения внутри блоков к полученному решению.
  - 4: Вернуть лучшее найденное решение в качестве результата выполнения алгоритма.
- 

Проведен вычислительный эксперимент с целью сравнения алгоритма 2 с алгоритмом  $LT_V^2$ . Было построено 4 серии тестовых примеров по 50 и 100 работ, обладающих различными свойствами, а также одна серия специальной блочной структуры, для которой решения алгоритма  $LT_V^2$  имеют относительное отклонение от нижней оценки целевой функции более 50%. Алгоритм 2 демонстрирует статистически значимо лучшие результаты, т.к. учитывает особенность структуры решений. Среднее относительное отклонение решений алгоритма 2 от нижней оценки на специальной серии составляет 7,8%, в то время как у алгоритма  $LT_V^2$  это значение равно 23,9%. На остальных сериях максимальное среднее относительное отклонение составляет 4,0% для алгоритма 2 и 8,27% для алгоритма  $LT_V^2$ . В тексте второй главы приводится подробное описание результатов эксперимента по каждой серии, а также зависимость качества решений от количества работ, выбранного параметра  $\alpha$  и других. Кроме этого представлено сравнение результатов алгоритма с результатами коммерческих решателей, таких как Gurobi и Baron.

Для задачи построения энергоэффективных расписаний выполнения работ на процессорах предложен алгоритм списочного типа, который по заданной перестановке находит оптимальное распределение энергии между работами для этой перестановки и строит расписание. В первую очередь алгоритм проводит

корректировку заданной перестановки  $\pi$  до допустимой с точки зрения моментов поступления и директивных сроков, а после этого корректирует моменты поступления и директивные сроки согласно этой перестановке. Полученная после корректировки задача составления расписаний имеет согласованные моменты поступления и директивные сроки, поэтому полиномиально разрешима [Yao F., 1995].

В вычислительном эксперименте сравниваются разные версии предложенного алгоритма с различными стратегиями построения перестановки. Тестирование проводилось на двух сериях: первая серия построена случайным образом, а структура второй серии позволяет заранее знать оптимальное решение. Для первой серии лидером среди всех алгоритмов является алгоритм, который упорядочивает работы по неубыванию моментов поступления. Эта стратегия выглядит естественной для случайной серии, т.к. моменты поступления, директивные сроки и объемы работ равномерно выбраны из одинаковых интервалов. Однако для второй серии лидером является алгоритм, который учитывает объемы и временные окна работ и упорядочивает работы по невозрастанию  $V_j/(d_j - r_j)$ . Это связано со спецификой построения серии.

Результаты данной главы опубликованы в [5; 6].

В **третьей главе** представлен генетический алгоритм (ГА) для решения рассматриваемых задач составления расписаний выполнения работ с учетом их взаимного влияния.

---

### Алгоритм 3 Адаптивная схема выбора оператора кроссинговера

---

- 1: Выбрать оператор кроссинговера  $a$ : с вероятностью  $\delta$  выбирается лучший оператор, с вероятностью  $(1 - \delta)$  оператор выбирается пропорционально его весу.
- 2: Применить оператор кроссинговера  $a$  к родительским особям.
- 3: Вычислить оценку выбранного оператора кроссинговера  $a$ :

$$\phi_a = \begin{cases} w_1, & \text{если найденное решение является новым лучшим} \\ & \text{найденным решением (среди всех перезапусков),} \\ w_2, & \text{если найденное решение лучше, чем лучшее найденное} \\ & \text{решение с момента последнего перезапуска,} \\ w_3, & \text{если найденное решение лучше как минимум одного из} \\ & \text{родителей,} \\ 0, & \text{иначе.} \end{cases}$$

- 4: Обновить вес оператора кроссинговера  $a$  по формуле:

$$\rho_a := \lambda \rho_a + (1 - \lambda) \phi_a.$$


---

Решения кодируются как перестановки на множестве работ. Условием завершения работы генетического алгоритма является достижение заданного количества вычислений значений целевой функции  $n_{max}$  или отведенного времени работы. В алгоритме используется классическое правило перезапуска: ГА перезапускается, как только номер текущей итерации становится равным номеру итерации, на которой была найдена лучшая особь, плюс заданное количество вычислений целевой функции  $n_{FC}$ . В качестве операторов кроссинговера были рассмотрены: одноточечный кроссинговер, циклический кроссинговер, порядковый кроссинговер, кроссинговер с частичным отображением. Также приближенно решается задача оптимальной рекомбинации, для этого реализованы оптимизированные и частично оптимизированные версии одноточечного кроссинговера, порядкового кроссинговера и кроссинговера с частичным отображением. В качестве операторов мутации были рассмотрены оператор сдвига и оператор обмена, а также реализовано две схемы применения оператора мутации: перемешивающий оператор мутации (scramble) на основе распределения Пуассона и мутация с тяжелыми хвостами (heavy-tailed) на основе степенного распределения. Для селекции реализованы такие операторы, как ранговая селекция и турнирная селекция.

На каждой итерации происходит выбор оператора кроссинговера: случайный оператор выбирается с вероятностью  $\varepsilon$ , а с вероятностью  $(1 - \varepsilon)$  применяется адаптивная схема 3. Здесь  $w_i$ ,  $i = 1, 2, 3$  и  $\delta$  – настраиваемые параметры,  $\lambda$  – так называемый параметр затухания, который контролирует, насколько чувствительны веса к изменениям в работе операторов.

Была построена оценка среднего числа итераций до первого достижения оптимума для задачи построения расписания для многоядерного процессора с учетом потребления шины данных и как следствие доказано, что предложенный алгоритм почти наверное сходится к оптимуму при числе итераций, стремящихся к бесконечности. Аналогичный результат имеет место для задачи построения энергоэффективных расписаний выполнения работ на процессорах.

В вычислительном эксперименте параметры алгоритма настраиваются с помощью пакета IRACE.

Для тестирования генетического алгоритма  $GA$  на задаче построения расписания для многоядерного процессора с учетом потребления шины данных были взяты серии тестовых примеров большой размерности (с числом работ равным 100) из первой главы. Сравнение проводилось с предложенным в первой главе алгоритмом списочного типа  $LT_p$ . Кроме критерия минимизации общего времени завершения всех работ ( $C_{max}$ ) был исследован критерий минимизации суммарного времени завершения работ ( $\sum_j C_j$ ). Результаты тестирования для разных типов частичного порядка на 100 работах представлены в таблице 1.

Результаты эксперимента показывают, что  $GA$  существенно превосходит алгоритм списочного типа  $LT_p$  при наличии нетривиальной структуры зависимостей между работами, особенно при критерии  $\sum_j C_j$ . При этом характер

Таблица 1 — Относительные отклонения решений  $LT_p$  от решений  $GA$ , %

	$C_{\max}$				$\sum_j C_j$			
	No-PO	B-PO	O-PO	R-PO	No-PO	B-PO	O-PO	R-PO
<b>min</b>	0.0	0.0	0.0	0.0	81.2	0.28	4.85	0.0
<b>avg</b>	0.0	13.42	4.85	0.0	95.74	8.64	27.3	0.0
<b>max</b>	0.0	24.89	17.62	0.0	120.98	23.5	70.73	0.0

Таблица 2 — Относительные отклонения решений алгоритмов  $GA$ ,  $LT_V^2$  и  $LI_V^2$  от нижней границы, %

	Серия 1			Специальная серия		
	$GA$	$LT_V^2$	$LI_V^2$	$GA$	$LT_V^2$	$LI_V^2$
<b>min</b>	0.8%	2.2%	1.7%	2.87%	51%	9.9%
<b>avg</b>	1.95%	8.33%	4.56%	3.88%	51.17%	11.9%
<b>max</b>	3.63%	16.4%	7.7%	4.54%	51.4%	17.1%

улучшений зависит от структуры частичного порядка. В тексте третьей главы приводится подробное описание результатов эксперимента для разных версий  $GA$  для серий тестовых примеров разной размерности, а также решения  $GA$  сравниваются с оптимальными решениями на примерах малой размерности.

Для тестирования алгоритма на задаче составления расписания выполнения распараллеливаемых работ на процессорах с ограничением на расход энергии тестовые примеры были сгенерированы способом, представленным во второй главе, а сравнение проводилось с алгоритмом списочного типа  $LT_V^2$  и алгоритмом с локальными улучшениями 2 ( $LI_V^2$ ).

В таблице 2 представлены результаты тестирования генетического алгоритма  $GA$  для серии 1 и серии специальной блочной структуры. Генетический алгоритм показывает статистически значимо лучшие результаты по сравнению с алгоритмами списочного типа, а максимальное относительное отклонение его решений от нижней границы не превышает 5% даже на примерах сложной серии. В тексте третьей главы приводится подробное описание результатов эксперимента для других версий генетического алгоритма, а также сравнение предложенного генетического алгоритма с современными коммерческими пакетами (Gurobi, Baron) на задачах малой размерности.

Тестирование алгоритма на задаче построения энергоэффективных расписаний выполнения работ на процессорах также проводилось на двух сериях тестовых примеров, способ генерации которых описан во второй главе. Сравнение проводилось с алгоритмом  $PLS - Many$ , представленным Ю.В. Захаровой в 2023 году. Результаты представлены в таблице 3, где  $GA_{opt}$  — это версия

Таблица 3 — Относительные отклонения решений алгоритмов  $GA$ ,  $GA_{орх}$  и  $PLS - Many$  от нижней границы, %

	Серия 1			Специальная серия	
	$GA$	$GA_{орх}$	$PLS - Many$	$GA$	$GA_{орх}$
<b>min</b>	0.00%	0.00%	0.0%	0.0%	0.0%
<b>avg</b>	0.1%	0.00%	0.9%	0.37%	0.0%
<b>max</b>	0.1%	0.04%	6.1%	11.22%	0.0%

генетического алгоритма с одноточечным оператором кроссинговера. Для первой серии статистически значимо лучшие результаты показал алгоритм  $GA_{орх}$ , а для специальной серии с известным оптимумом алгоритм  $GA$  только в одном примере из серии не нашел оптимальное решение. В тексте третьей главы приводится подробное описание результатов эксперимента для других версий генетического алгоритма.

Результаты данной главы опубликованы в [2; 4].

В четвертой главе описаны разработанные в рамках диссертационного исследования программы для ЭВМ, объединённые в комплекс программ и позволяющие на основе анализа свойств предложенных математических моделей, разработанных алгоритмов приближенного решения и численных методов составлять расписания выполнения работ в компьютерных системах с учетом их взаимного влияния, а также оценивать полученные расписания в реальном эксперименте. Описано практическое применение разработанного комплекса на двух задачах из реальной практики:

1. Задача размещения виртуальных машин на серверах, представляющая интерес для компаний из сферы облачных технологий. Предложенная методика позволяет строить оценки числа задействованных серверов как снизу, так и сверху. Более того, учёт характеристик выполняемых работ позволяет получать решения более высокого качества по сравнению с классическими алгоритмами, такими как First Fit. Результаты предложенного алгоритма на открытом наборе данных, предоставленном компанией Хуавей<sup>1</sup>, показывают отклонение от нижней границы 3,5%.

2. Задача планирования выполнения работ на ядрах процессоров Intel, актуальная для производителей вычислительных систем. В качестве работ выступают процедуры из библиотеки oneMKL, а сравнение проводится с широко используемым в современном программном обеспечении планировщиком oneTBV. Среднее преимущество решений модели ЧЦЛП, найденных пакетом прикладных программ CPLEX, по сравнению с решениями, полученными с помощью библиотеки oneTBV, составляет 7,2%, а по сравнению с предложенным жадным алгоритмом, обладающим малой трудоемкостью, – 3,31%.

Созданный комплекс программ включает следующие библиотеки:

<sup>1</sup><https://vmagent.readthedocs.io/en/latest/simulator/dataset.html>

- библиотеку классов решателя Heuristic, реализующую методы эволюционного программирования и конструктивные алгоритмы;
- библиотеку для запуска моделей математического программирования Solver;
- комплекс программ для реализации расписаний для задачи планирования работ на ядрах процессора в реальном эксперименте.

Для каждой из перечисленных выше библиотек описаны функциональные возможности, а также представлены форматы входных и выходных данных.

Результаты данной главы отражены в акте о внедрении и трех свидетельствах о государственной регистрации программ для ЭВМ.

В **заключении** приведены основные результаты работы, которые заключаются в следующем:

1. Для задач составления расписаний с учетом ограничения на потребление общего ресурса и свойства распараллеливания:

- исследованы свойства оптимизационных моделей: предложен способ формирования конфигураций работ, позволивший построить модели частично целочисленного программирования с непрерывным представлением времени; выявлена блочная структура решений задачи на двух процессорах с суммарным ограничением на потребление энергии; доказана NP-трудность в случае однопроцессорных работ и учета пропускной способности шины данных и в случае распараллеливаемых работ с суммарным ограничением на потребление энергии.

- разработаны конструктивные алгоритмы списочного типа с поэтапным распределением ресурса при назначении очередной работы в расписание; алгоритм жадного типа с проблемно-ориентированной стратегией назначения работ, где учитывается текущий уровень потребления ресурса при выборе очередной работы; алгоритмы имеют малую трудоемкость.

- предложен метод локальных улучшений, учитывающий блочную структуру решений и использующий проблемно-ориентированные окрестности, для задачи составления расписаний с возможностью распараллеливания работ и ограничением на расход энергии.

- разработан адаптивный эволюционный алгоритм с оптимизированными операторами, где решения кодируются как перестановки, по которым допустимые расписания восстанавливаются с помощью оригинальных предложенных процедур декодирования, учитывающих специфику постановок задач; размерность оптимизационных подзадач в операторах настраивается адаптивно методами обучения с подкреплением.

- алгоритмы демонстрируют конкурентоспособные результаты на примерах различной структуры, построенных по аналогии с примерами из реальной практики.

2. Разработан комплекс программ, включающий в себя предложенные математические модели, конструктивные алгоритмы и метаэвристики для решения задач с учетом ограничения на потребление общего ресурса и свойства распараллеливания. При разработке применялись современные компьютерные технологии и эффективные методы реализации программ.

3. Экспериментальное исследование подтвердило возможность и эффективность применения предложенного математического аппарата, численных методов и комплекса программ для решения практических задач, возникающих в компьютерных системах:

- планирования процедур из библиотеки oneMKL на ядрах процессора Intel с учетом их взаимного влияния;

- распределения виртуальных машин по серверам с NUMA-архитектурой.

Разработанные в диссертации конструктивные алгоритмы и выявленные свойства расписаний применяются в ООО «Техкомпания Хуавей» при анализе системы планирования работ.

Дальнейшее развитие предложенных в работе подходов и инструментов может быть связано с их обобщением на случай заданного набора ресурсов, характерных для многопроцессорных компьютерных систем. Кроме того, указанные методы могут быть адаптированы для задач построения производственных расписаний, обладающих аналогичной структурой: наличием частичного порядка между операциями, ограничениями на использование ресурсов и возможностью обработки одной детали несколькими машинами.

## СПИСОК ОСНОВНЫХ ПУБЛИКАЦИЙ

**Научные публикации в изданиях из перечня ВАК и приравненных к ним:**

1. Еремеев, А. В. Построение расписания для многоядерного процессора с учетом взаимного влияния работ / А. В. Еремеев, М. Ю. Сахно. – DOI: 10.26089/NumMet.v24r108 // Вычислительные методы и программирование. – 2023. – Т. 24, № 1. – С. 115–126.

2. Сахно, М. Ю. Адаптивный генетический алгоритм с оптимальной рекомбинацией для задачи составления расписаний с учетом расхода энергии / М. Ю. Сахно. – DOI: 10.15372/SJNM20250307 // Сибирский журнал вычислительной математики. – 2025. – Т. 28, № 3. – С. 327–346.

3. Multi-core processor scheduling with respect to data bus bandwidth / A. V. Ereemeev, A. A. Malakhov, M. A. Sakhno, M. Y. Sosnovskaya. – DOI: 10.1007/978-3-030-65739-0\_5 // Optimization and Applications: 11th International Conference, OPTIMA 2020, Moscow, Russia, September 28 – October 2, 2020, Proceedings. Vol. 12422 LNCS / ed. by N. Olenev [et al.]. – Cham : Springer, 2020. – P. 55–69.

4. Zakharova, Y. V. Adaptive genetic algorithm with optimized operators for scheduling in computer systems / Y. V. Zakharova, M. Y. Sakhno. – DOI: 10.1007/978-3-031-57808-3\_23 // Intelligent Information Processing XII: 13th IFIP TC 12 International Conference, IIP 2024, Shenzhen, China, May 3–6, 2024, Proceedings. Vol. 703 IFIPAICT / ed. by Z. Shi [et al.]. – Cham : Springer, 2024. – P. 317–328.

5. Zakharova, Y. V. Complexity and heuristic algorithms for speed scaling scheduling of parallel jobs with energy constraint / Y. V. Zakharova, M. Y. Sakhno. – DOI: 10.1016/j.cam.2024.116254 // Journal of Computational and Applied Mathematics. – 2025. – Vol. 457. – Article ID: 116254.

6. Zakharova, Y. V. Heuristics with local improvements for two-processor scheduling problem with energy constraint and parallelization / Y. V. Zakharova, M. Y. Sakhno. – DOI: 10.1007/978-3-031-81241-5\_17 // Numerical Computations: Theory and Algorithms: 4th International Conference, NUMTA 2023, Pizzo Calabro, Italy, June 14–20, 2023, Revised Selected Papers, Part I. Vol. 14476 LNCS / ed. by Y. D. Sergeyev [et al.]. – Cham : Springer, 2025. – P. 241–256.

## **В прочих изданиях**

7. Сахно, М. Ю. Алгоритм списочного типа для размещения виртуальных машин на сервера с учетом NUMA-архитектуры / М. Ю. Сахно // Прикладная математика и фундаментальная информатика (ПМиФИ). – 2025. – Т. 12, № 3. – С. 9–14.

8. Сахно, М. Экспериментальное исследование методов составления расписаний для многоядерных процессоров / М. Сахно // ФМХ ОмГУ 2022: сб. статей X региональной конф. магистрантов, аспирантов и молодых ученых по физике, математике и химии (Омск, 6–19 июня 2022) / под ред. Ю. В. Захарова, Г. М. Серопян. – Омск : ОмГУ, 2022. – С. 18–21.

## **Свидетельства о государственной регистрации программ для ЭВМ:**

9. Свидетельство о государственной регистрации программы для ЭВМ № 2025682254 Российская Федерация. Программа для реализации расписаний при планировании работ на ядрах процессора с учетом их взаимного влияния : № 2025681187 : заявл. 13.08.2025 : опубл. (зарег.) 21.08.2025 / М. Ю. Сахно ; заявитель Сахно Мария Юрьевна. – 1 с.

10. Свидетельство о государственной регистрации программы для ЭВМ № 2025682936 Российская Федерация. Конструктивные эвристические алгоритмы поиска решений для задач составления расписаний в многопроцессорных системах с ресурсными ограничениями : № 2025680891 : заявл. 13.08.2025 : опубл. (зарег.) 28.08.2025 / М. Ю. Сахно ; заявитель Сахно Мария Юрьевна. – 1 с.

11. Свидетельство о государственной регистрации программы для ЭВМ № 2025683991 Российская Федерация. Программа для решения задачи составления расписаний в компьютерных системах с ресурсными ограничениями на основе генетического алгоритма : № 2025680862 : заявл. 13.08.2025 : опубл. (зарег.) 10.09.2025 / М. Ю. Сахно ; заявитель Сахно Мария Юрьевна. – 1 с.