

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

**Разработка системы автоматизированного составления расписания
научной конференции на основе генетического алгоритма**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Акишиной Полины Дмитриевны

Научный руководитель:

ст. преподаватель

Портенко М. С.

подпись, дата

Зав. кафедрой:

к.ф.-м.н., доцент

Огнева М. В.

подпись, дата

Саратов 2026

ВВЕДЕНИЕ

Актуальность темы. Создание расписания для научной конференции является сложной организационной задачей, для которой требуется учесть целый ряд взаимосвязанных факторов, такие как тематика докладов, их продолжительность, количество параллельных секций, вместимость залов и общие временные рамки мероприятия. Решение этой задачи вручную может оказаться слишком трудоемким, а также не всегда приведет к оптимальным результатам, могут возникнуть пересечения тем, неравномерная загрузка секций и общее снижение качества программы. Несмотря на то, что существуют работы, посвященные автоматизации составления расписаний, проблема разработки специализированных решений для научных конференций, сочетающих алгоритмическую генерацию с удобными инструментами для ручной корректировки, остается актуальной. В свою очередь, создание такой системы позволит перевести процесс из рутинного подбора в область контролируемой оптимизации.

Цель бакалаврской работы – разработка и практическая реализация системы автоматизированного планирования программы научной конференции, включающей модуль генерации расписания на основе генетического алгоритма и интуитивный веб-интерфейс для его визуализации и ручного редактирования.

Поставленная цель определила **следующие задачи:**

1. Провести анализ современных научных и методических источников, посвященных задаче автоматического составления расписаний, и существующих программных решений.
2. Изучить и сравнить различные классы алгоритмов, применяемых для решения задач комбинаторной оптимизации.
3. Обосновать выбор генетического алгоритма в качестве базового метода для решения поставленной задачи.

4. Формализовать задачу составления расписания конференции, определить множество входных данных, систему жестких и мягких ограничений, а также целевую функцию.
5. Разработать детальную архитектуру системы, включающую backend-модуль, frontend-интерфейс и модуль алгоритмической генерации.
6. Выбрать и обосновать выбор стека технологий для реализации каждого компонента системы.
7. Реализовать генетический алгоритм, включая разработку схемы кодирования особи, функций инициализации популяции, скрещивания, мутации, кроссинговера и селекции.
8. Реализовать функцию приспособленности, учитывающую ключевые критерии качества расписания и подобрать весовые коэффициенты.
9. Разработать полнофункциональный пользовательский интерфейс.
10. Провести комплексное тестирование системы.

Методологические основы Методологические основы задачи автоматизированного составления расписания научной конференции представлены в работах следующих авторов: Балтак С.В., Сотсков Ю.Н., Устинов И.А., Набродова И.Н., Завьялов А.М., Новиков А.В., Панченко Т.В., Еремеев А.В., Гладков Л.А., Курейчик В.В., Курейчик В.М., Мицель А.А., Cicirello V.A. и др.

Теоретическая значимость бакалаврской работы заключается в формализации задачи составления расписания научной конференции как многокритериальной оптимизационной задачи с системой жестких и мягких ограничений, разработке целевой функции на основе взвешенных штрафов и адаптации генетического алгоритма с кодированием перестановок для данной предметной области.

Практическая значимость состоит в создании полнофункционального веб-приложения, включающего модуль автоматической генерации расписания на основе генетического алгоритма, визуальный редактор с поддержкой drag-and-drop для ручной корректировки,

подсистему генерации PDF-программы конференции, а также комплекс эксплуатационных механизмов (обновление тезисов, удаление заявок, массовое скачивание файлов, управление учётными записями).

Структура и объём работы. Бакалаврская работа состоит из введения, двух разделов, заключения, списка использованных источников и четырёх приложений. Общий объём работы – 103 страницы, из них 74 страницы – основное содержание, включая 11 рисунков, список использованных источников информации – 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Теоретическая часть» посвящен анализу существующих подходов к автоматизации составления расписаний, математической формализации задачи и описанию генетического алгоритма.

В разделе выполнен обзор научных источников и программных продуктов (EasyChair, ConfTool, Indico, OpenConf), показано, что существующие системы не предоставляют встроенного оптимизационного ядра для автоматической генерации расписания. Рассмотрены различные классы алгоритмов: точные методы (раскраска графа), метаэвристики (генетический алгоритм, алгоритм имитации отжига), методы логического программирования в ограничениях. Обоснован выбор генетического алгоритма как наиболее гибкого и эффективного для данной задачи.

Проведена математическая постановка задачи: введены множества докладов, аудиторий, временных слотов и тематических секций, сформулированы жесткие ограничения (единственность размещения, занятость аудитории, непрерывность доклада) и мягкие ограничения (тематические пересечения, равномерность загрузки, удобная последовательность). Разработана целевая функция как взвешенная сумма штрафов, включая штраф за тематические пересечения, неравномерную загрузку аудиторий и неудобную последовательность докладов. Для

определения степени схожести тематик использована матрица, построенная на основе метрик классификации.

Описана структура генетического алгоритма: кодирование решений в виде перестановок идентификаторов докладов, операторы селекции (турнирный отбор), кроссинговера (порядковый кроссовер NWOX), мутации (swap и insertion). Обоснован выбор параметров алгоритма: размер популяции 100 особей, вероятность кроссинговера 0,85, адаптивная вероятность мутации (от 2% до 0,5%), элитизм 2 лучшие особи, критерий остановки – 1000 поколений или 50 поколений без улучшения лучшего решения. Рассмотрена проблема преждевременной сходимости и предложены методы её предотвращения.

Выводы по первому разделу: задача составления расписания научной конференции относится к классу NP-трудных комбинаторных задач; генетический алгоритм с кодированием перестановок является наиболее подходящим методом решения; разработанная математическая модель позволяет формализовать все ключевые требования к расписанию.

Второй раздел «Практическая часть» посвящен программной реализации генетического алгоритма, интеграции в веб-приложение, разработке пользовательского интерфейса, созданию PDF-генератора и эксплуатационных механизмов.

Самостоятельно разработан модуль генетического алгоритма, включающий:

- класс Individual для представления особи (хромосома, значение приспособленности, расписание)
- функцию decode, преобразующую перестановку идентификаторов докладов в календарное расписание с учётом всех жёстких ограничений (временные слоты, кофе-брейки, эксклюзивное время для пленарных и стендовых докладов, занятость комнат). Функция автоматически добавляет системные блоки (регистрация, открытие, закрытие, стендовая секция)

- функцию приспособленности `calculateFitness`, которая вычисляет суммарный штраф на основе пяти компонентов: `themePenalty` (тематические пересечения), `loadPenalty` (равномерность загрузки залов), `plenaryPenalty` (нарушения для пленарных докладов), `exclusiveTimePenalty` (пересечения с эксклюзивным временем), `dayLoadPenalty` (равномерность по дням). Весовые коэффициенты подобраны эмпирически
- эволюционный цикл `runGA` с турнирной селекцией (размер турнира 3), порядковым кроссовером `NWOX`, адаптивной вероятностью мутации (линейное уменьшение от 2% до 0,5%), элитизмом (2 лучшие особи) и критерием стагнации (50 поколений)

Реализована `backend`-интеграция модуля расписания через маршруты `Express.js`: `POST /schedule/generate` (запуск генерации), `GET /schedule` (просмотр расписания с учётом прав доступа), `POST /schedule/visibility` (переключение видимости), `GET /schedule/edit` и `POST /schedule/edit` (ручное редактирование с сохранением). Серверная часть использует `MongoDB` для хранения расписания и докладов.

Разработан `frontend`-интерфейс, состоящий из двух основных страниц: просмотра расписания (`views/schedule.hbs`) и редактора с поддержкой `drag-and-drop` (`views/schedule-edit.hbs`).

Страница просмотра расписания отображает программу в виде набора таблиц. При рендеринге сервер формирует структуру `dayGrids`: для каждого дня определяется уникальный набор временных слотов (на основе событий и кофе-брейков), затем для каждого слота выводятся все доклады и системные блоки, назначенные на это время. Кофе-брейки отображаются отдельными строками с заливкой фона, причём они добавляются только при наличии докладов после времени перерыва. Для пользователей из оргкомитета на странице дополнительно выводятся кнопки управления видимостью расписания, перехода к редактору и генерации PDF-программы.

Страница редактора предоставляет оргкомитету возможность вносить ручные правки без повторного запуска генетического алгоритма. Каждое

событие представлено в виде карточки, содержащей время начала и окончания, длительность, название, докладчика, комнату, а также кнопки удаления и редактирования. Реализована полноценная поддержка drag-and-drop: при начале перетаскивания карточка становится полупрозрачной, допустимые ячейки подсвечиваются, а при сбросе автоматически пересчитывается время окончания события (на основе исходной длительности). Для редактирования и создания событий используется модальное окно, в котором можно изменить день, время начала, длительность, название, докладчика и комнату. Все изменения накапливаются в клиентском объекте `scheduleData.items`, а сохранение происходит асинхронно через `fetch`-запрос на сервер, который обновляет документ в MongoDB. После успешного сохранения страница перенаправляется на просмотр обновлённого расписания. Такой подход сочетает удобство визуального перемещения с мощными средствами точной настройки, обеспечивая гибкость при финальной доработке программы конференции.

Создан модуль генерации PDF-программы конференции (`utils/programPdf.js`) с использованием библиотеки `pdfkit`. Программа включает титульный лист, почасовые таблицы расписания для каждого дня конференции (с автоматическим разбиением на страницы), детализированные списки докладов по секциям и отдельный раздел стендовой секции. Реализован поиск системного шрифта с поддержкой кириллицы.

Разработаны эксплуатационные доработки:

- обновление версии тезисов (для участников и оргкомитета)
- удаление поданной заявки с очисткой файлов
- смена секции доклада (для руководителя секции и администратора)
- подача доклада оргкомитетом от лица участника (минуя временные ограничения, со статусом «Принят»)
- массовое скачивание тезисов в ZIP-архиве

- управление учётными записями участников (массовое удаление за год и точечное удаление)

Создан текстовый конфигурационный файл info.txt, в котором хранятся ежегодно изменяемые параметры: даты проведения конференции, перечень секций, список доступных залов, ссылки на трансляции, состав программного комитета. Файл имеет простую текстовую структуру с комментариями-разделителями и доступен для редактирования через веб-интерфейс администратора без остановки приложения и без привлечения разработчика. Парсинг выполняется как на сервере (Node.js, синхронное чтение), так и на клиенте (вставка содержимого в скрытый элемент и последующий построчный разбор). Такой подход позволяет оргкомитету самостоятельно менять любые ключевые параметры конференции, избегая сложностей с базами данных или перекомпиляцией кода.

Проведено тестирование и анализ сходимости генетического алгоритма на реальных данных (56 докладов, 4 дня, 2 зала). Экспериментальные запуски показали: лучшее значение целевой функции достигается за 2,4 секунды, алгоритм стабильно находит решения с суммарным штрафом в диапазоне 282-362 единиц, критерий стагнации срабатывает на 101-м поколении. График сходимости демонстрирует резкое улучшение на первых поколениях и последующую стабилизацию.

Выводы по второму разделу: разработанный генетический алгоритм успешно решает задачу составления расписания, все жёсткие ограничения соблюдаются, качество расписания оценивается комплексной функцией штрафов. Веб-интерфейс обеспечивает удобный просмотр и мощные средства ручной корректировки. Подсистема генерации PDF автоматически создаёт официальную программу конференции. Эксплуатационные доработки покрывают полный жизненный цикл заявки и существенно сокращают рутинные операции оргкомитета.

ЗАКЛЮЧЕНИЕ

В соответствии с поставленными задачами, в результате выполнения выпускной квалификационной работы была разработана и практически реализована система автоматизированного составления расписания научной конференции. Центральным элементом системы является генетический алгоритм, использующий кодирование решений в виде перестановок, турнирную селекцию, адаптивную вероятность мутации и порядковый кроссовер NWOX. Реализованная функция приспособленности учитывает штрафы за тематические пересечения, неравномерную загрузку залов, неудобную последовательность докладов и другие критерии качества. Экспериментальные запуски на реальных данных (56 докладов) подтвердили сходимость алгоритма и его эффективность, лучшее решение находится в среднем за 2,4 секунды, а значение целевой функции стабилизируется в узком диапазоне.

Помимо алгоритмического ядра, создан полнофункциональный веб-интерфейс для оргкомитета и участников, включающий страницы просмотра расписания с табличным отображением по дням и визуальный редактор с поддержкой drag-and-drop для ручной корректировки. Реализована автоматическая генерация PDF-программы конференции с титульным листом, почасовыми таблицами и детальными списками докладов по секциям. Для обеспечения эксплуатационной гибкости разработаны механизмы обновления тезисов, удаления заявок, массового скачивания файлов и управления учётными записями.

Таким образом, цель работы достигнута, а полученные результаты подтверждают практическую применимость системы для автоматизации планирования программ научных мероприятий.

Основные источники информации:

1. Балтак, С. В., Сотсков, Ю. Н. Построение расписаний учебных занятий на основе раскраски вершин графа // Информатика. – 2006. – №3. – С. 58-69.

2. Завьялов, А. М., Новиков, А. В. Автоматизация задачи составления учебного расписания // Системный анализ в науке и образовании. – 2009. – Выпуск №1. – С. 1-20.
3. Устинов, И. А., Набродова, И. Н. Анализ существующих алгоритмов для составления расписания занятий // Известия Тульского государственного университета. Технические науки. – 2012. – Вып. 9. – С. 109-112.
4. Панченко, Т. В. Генетические алгоритмы : учебно-методическое пособие / под ред. Ю. Ю. Тарасевича. – Астрахань : Издательский дом «Астраханский университет», 2007. – 87 с.
5. Гладков, Л. А., Курейчик, В. В., Курейчик, В. М. Генетические алгоритмы / Под ред. В. М. Курейчика. – 2-е изд., исправл. и доп. – М.: ФИЗМАТЛИТ, 2010. – 368 с.
6. Мицель, А. А. Эвристические методы оптимизации : учебное пособие. – Томск: Томск. гос. ун-т систем упр. и радиоэлектроники, 2024. – 73 с.
7. Cicirello, V. A. Non-Wrapping Order Crossover: An Order Preserving Crossover Operator that Respects Absolute Position // University of York, GECCO. – 2006. – С. 1125-1131.
8. Горовых, И. И., Горовых, С. Н. Гибридный подход к решению NP-сложных задач настройки производства: объединение метода ветвей и границ с генетическим алгоритмом // Молодой исследователь Дона. – 2025. – 9(1). – С. 10-19.