

УДК 004.4

РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ СОСТАВЛЕНИЯ РАСПИСАНИЯ ОБРАЗОВАТЕЛЬНОЙ ОРГАНИЗАЦИИ

Цебренько К.Н.

к.т.н., доцент,

*Академия маркетинга и социально-информационных технологий-ИМСИТ (г.
Краснодар),*

г. Краснодар, Российская Федерация

Аннотация: В статье рассмотрена задача автоматизации процесса формирования учебного расписания в образовательных организациях. Представлена разработка веб-системы на базе фреймворка Django. Для генерации расписания использована комбинация «жадного» и «генетического» алгоритмов. Описаны этапы проектирования системы, выбор технологического стека, реализация алгоритмов и интерфейс пользователя. Проведено тестирование системы, включая экспериментальное сравнение времени работы алгоритмов до и после оптимизации кода. Разработанная система способствует повышению эффективности управления учебным процессом и улучшению условий для обучающихся и преподавателей.

Ключевые слова: автоматическое расписание, образовательная организация, веб-система, Django, жадный алгоритм, генетический алгоритм, оптимизация расписания, SQLite, производительность алгоритмов, цифровизация образования.

DEVELOPMENT OF AN AUTOMATED SYSTEM FOR PREPARING A SCHEDULE FOR AN EDUCATIONAL ORGANIZATION

Tsebrenko K.N.

PhD, Associate professor,

Academy of Marketing and Social information technologies – IMSIT,

Krasnodar, Russian Federation,

Abstract: The article considers the task of automating the process of forming an academic schedule in educational organizations. The development of a web system based on the Django framework is presented. A combination of "greedy" and "genetic" algorithms is used to generate the schedule. The stages of system design, the choice of the technology stack, the implementation of algorithms and the user interface are described. The system is tested, including an experimental comparison of the running time of the algorithms before and after code optimization. The developed system helps to increase the efficiency of educational process management and improve conditions for students and teachers.

Keywords: automatic schedule, educational organization, web system, Django, greedy algorithm, genetic algorithm, schedule optimization, SQLite, algorithm performance, digitalization of education.

Разработка систем автоматического расписания занятий образовательной организации представляет собой актуальное и важное направление в цифровизации образовательных процессов. Реализация сервисов образовательных организаций нередко выполняется с использованием веб-технологий [1]. Веб-системы обладают возможностями быстрого роста и расширяемости обновлений без необходимости установки специализированного программного обеспечения на устройства пользователей. Это способствует оперативному внедрению изменений и обновлений системы. Данное направление крайне актуально в контексте развития собственного программного обеспечения цифровых сервисов образовательной организации. Разработка программного обеспечения, позволяющего формировать расписание в автоматическом режиме, позволит создать функционал, полностью соответствующей потребностям учебного заведения [2]. В работе [2] описана концепция разработки автоматизированной системы формирования расписания в образовательной организации с использованием веб-технологий. На ее основе построена объектная модель системы [3], положенная в основу проектной документации на разработку программного обеспечения системы [4].

В процессе разработки выполнен подбор технологического стека. Выбор фреймворка может в значительной степени повлиять на успех проекта, поэтому важно тщательно изучить все доступные варианты и выбрать тот, который будет наилучшим образом соответствовать целям. Предоставляя готовые инструменты, библиотеки и структуры, фреймворки помогают сосредоточиться на бизнес-логике приложения, минимизируя написание повторяющегося кода.

Django – высокоуровневый фреймворк, ускоряющий разработку веб-приложений за счет встроенных инструментов для управления данными, аутентификации и администрирования. Он следует принципам «DRY» (Don't Repeat Yourself) и конвенции перед конфигурацией. Django широко используется для создания крупных и сложных веб-приложений, включая социальные сети, электронную коммерцию и CRM-системы [5].

Сравнение фреймворков по данным критериям представлено в таблице 1.

Таблица 1 – Сравнение фреймворков

Критерии	Django	Flask	Pyramid	CherryPy
Уровень абстракции	высокий	низкий	средний	низкий
Скорость разработки	высокая	средняя	средняя	высокая
Сообщество и экосистема	обширное	обширное	среднее	малое
Масштабируемость	отличная	высокая	высокая	средняя
Безопасность	высокая	высокая	высокая	средняя
Функциональность	обширная	обширная	средняя	средняя

Благодаря огромному сообществу разработчиков, активно поддерживаемой документации и множеству сторонних библиотек и расширений, Django обеспечивает быструю и эффективную разработку и делает возможным рост и расширение практически неограниченным.

Достоинства Django для разработки системы автоматического расписания: ORM (Object-Relational Mapping), упрощающий работу с базами данных и позволяющий легко создавать сложные запросы, встроенная админ-панель, позволяющая управлять данными в приложении без необходимости писать

дополнительный код, масштабируемость, что делает его отличным выбором для разработки крупных и сложных систем, наличие документации и поддержки для самостоятельного изучения, большой набор библиотек, ускоряющих разработку системы автоматического расписания.

Система автоматического формирования расписания для образовательной организации разработана на Django и предназначена для упрощения процесса создания, управления и оптимизации учебного расписания. Она использует комбинацию жадного и генетического алгоритмов для эффективного распределения занятий между дисциплинами, преподавателями, аудиториями, группами [2] и включает интерактивные функции для ручного управления расписанием.

Система использует SQLite для хранения данных. Структура базы данных включает таблицы для дисциплин, преподавателей, аудиторий, групп и расписания. Django ORM используется для определения моделей и взаимодействия с базой данных. Основные модели включают Discipline, Teacher, Audience, Group и Schedule. Django views используются для обработки запросов и отображения данных. Основные представления включают создание расписания, просмотр списка расписаний и обновление записей. Django templates используются для отображения данных пользователю. Основные шаблоны включают форму для создания расписания и таблицу для просмотра расписания. Жадный алгоритм используется для быстрого создания первоначального расписания. Алгоритм работает следующим образом: занятия сортируются по приоритету или другим критериям, например, по сложности или времени проведения. Алгоритм последовательно назначает занятия в доступные аудитории, начиная с первой по списку. Проверяются доступность аудитории, преподавателя и группы на указанное время. Если обнаружен конфликт, алгоритм пытается найти другое доступное время или аудиторию для занятия. Если это невозможно, занятие остается неназначенным для последующей обработки.

Генетический алгоритм используется для оптимизации начального расписания. Дневник науки | www.dnevniknauki.ru | СМН ЭЛ № ФС 77-68405 ISSN 2541-8327

расписания, созданного жадным алгоритмом. Алгоритм работает следующим образом: создается начальная популяция расписаний на основе первоначального расписания. Каждое расписание оценивается по нескольким критериям, включая количество конфликтов, сбалансированность нагрузки и предпочтения пользователей. Выбираются лучшие расписания для создания нового поколения. Происходит обмен частями расписания между выбранными расписаниями для создания новых комбинаций. Вносятся случайные изменения в расписания для увеличения разнообразия. Процесс повторяется до достижения оптимального расписания или до достижения определенного количества итераций.

Система предоставляет возможность вручную управлять расписанием через интерфейс Django admin и интерактивные представления. Основные функции включают: просмотр расписания через интерфейс; редактирование расписания, где пользователи могут изменять время, аудиторию и преподавателя для занятий; удаление и добавление занятий через форму.

Система проверяет расписание на наличие конфликтов и других проблем: проверяется, что аудитория не занята более чем одной группой одновременно, что преподаватель не назначен на несколько занятий одновременно, и что группа не имеет пересекающихся занятий. Если система обнаруживает конфликты, она предлагает пользователям возможные решения или автоматические исправления. Система автоматического формирования расписания для образовательной организации на Django предоставляет инструмент для эффективного создания и управления учебным расписанием. Использование жадного и генетического алгоритмов позволяет автоматизировать процесс и оптимизировать результаты, обеспечивая при этом гибкость и интерактивность для пользователей [2].

Проведен анализ работы алгоритма с помощью блок-схемы. Описаны генетические алгоритмы, жадный алгоритм и интерактивный подход. В результате составлена блок-схема работы данного алгоритма, что позволяет легко понять последовательность действий и принцип работы алгоритма. После этого проведено описание и выбор подходящей IDE, в результате чего для Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

разработки системы автоматического формирования расписания выбран PyCharm. После этого описана работа системы автоматического формирования расписания для образовательной организации.

Создание базы данных для проекта Django выполнено с помощью «SQLite» и «SQLite Studio». Осуществлена разработка «Жадного алгоритма», полностью описан код и поэтапно описаны все функции, выполняемые этим кодом. Также разработан «Генетический алгоритм».

Для разработки интерфейса был выбран онлайн конструктор «Ninjamock», за простоту синтаксиса и удобный графический инструментарий. Затем выполнен перенос интерфейса с конструктора на фреймворк Django путём экспорта в HTML. На рисунке 1 представлен макет расписания групп с некоторыми возможностями. Слева страницы приставлено меню, в котором есть некоторые функции такие как «Выбор группы» группа выбирается либо из списка групп, либо через поиск, после чего для отображения расписания выбранной группы необходимо нажать кнопку «Отобразить». Справа от выбора групп находятся Функции, которые называются следующим образом: «Редактировать», «Сохранить», «Экспорт», «Импорт», «Печать» и «Удалить». Также переход по страницам с расписанием преподавателей и аудиторий (рисунок 2).

Пары	1-я	2-я	3-я	4-я	5-я	6-я	7-я
Время	08.00-09.30	09.40-11.10	11.30-13.00	13.10-14.40	14.50-16.20	16.30-18.00	18.10-19.40
Пнд							
Втр							
Срд							
Чтв							
Птн							
Сбт							

Рис. 1 – Расписание групп

Пары	1-я	2-я	3-я	4-я	5-я	6-я	7-я
Время	08:00-09:30	09:40-11:10	11:30-13:00	13:10-14:40	14:50-16:20	16:30-18:00	18:10-19:40
Пон							
Втр							
Сре							
Чет							
Пят							
Суб							

Рис. 2 – Расписание аудиторий

Функция «Редактировать» позволяет соответственно редактировать содержимое расписания для групп, а также список групп, аналогичный функционал на страницах с аудиториями и преподавателями. Функция «Сохранить» отвечает за сохранение изменений. Функция «Экспорт» отвечает за экспорт расписания, формат которых можно выбрать самостоятельно из

предложенных. Функция «Импорт» отвечает за импорт расписания из заранее закреплённых форматов файлов. Функция «Печать» отправляет расписание на печать. Функция «Удалить» очищает расписание выбранной группы.

Также на странице с каждым расписанием представлены алгоритмы для возможности автоматического формирования расписания с помощью «Жадного» или «Генетического» алгоритмов, которые можно применить, поставив «галочку» напротив нужного алгоритма, либо напротив обоих алгоритмов. Для формирования расписания в автоматическом режиме после выбора советующих алгоритмов необходимо нажать кнопку «Сформировать», если результат устраивает, то можно нажать кнопку «Сохранить», а если же нет можно повторно сформировать расписание или же отменить изменения нажав кнопку «Отменить».

Проведено тестирование системы с описанием функционала, в результате которого, можно сделать вывод о правильной работе алгоритмов, также проведён эксперимент по использованию алгоритмов. На рисунке 3 показано сгенерированное расписание с помощью выбранного алгоритма. Для просмотра расписания другой группы необходимо выбрать группу для отображения с помощью выпадающего списка и нажать кнопку «Отобразить расписание».

За создание начального расписания отвечает «Жадный алгоритм», а за его оптимизацию отвечает «Генетический алгоритм». После построения расписания для его оптимизации выставляем ограничения как показано на рисунке 4. После этого также выбираем алгоритм. Генетический алгоритм начнёт оптимизацию расписания с учётом ограничений.

Итог оптимизации генетического алгоритма для одной из групп показан на рисунке 5. Итоговое расписание соответствует заданным ограничениям из чего можно сделать вывод что алгоритмы работают корректно.

Расписание

Установить ограничения Оптимизации

Выберите группу:

Все группы

Выберите алгоритм:

Жадный алгоритм

Сгенерировать расписание

Выберите группу для отображения:

Группа А

Отобразить расписание

Пары	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
08:00-09:30	Математика Иван Иванов Аудитория 1	География Николай Николаев Аудитория 1	Математика Иван Иванов Аудитория 1	География Николай Николаев Аудитория 2	Химия Сергей Сергеев Аудитория 1	География Николай Николаев Аудитория 3
09:40-11:10	Химия Сергей Сергеев Аудитория 1	Физика Петр Петров Аудитория 1	География Николай Николаев Аудитория 2	Биология Дмитрий Дмитриев Аудитория 3	История Алексей Алексеев Аудитория 1	Литература Андрей Андреев Аудитория 3
11:30-13:00	Физика Петр Петров Аудитория 1	Литература Андрей Андреев Аудитория 1	Биология Дмитрий Дмитриев Аудитория 1	Литература Андрей Андреев Аудитория 1	Физика Петр Петров Аудитория 1	Математика Иван Иванов Аудитория 7
13:10-14:40	История Алексей Алексеев Аудитория 1	История Алексей Алексеев Аудитория 1	Химия Сергей Сергеев Аудитория 1	Химия Сергей Сергеев Аудитория 2	Математика Иван Иванов Аудитория 3	Физика Петр Петров Аудитория 3
14:50-16:20	Биология Дмитрий Дмитриев Аудитория 1	Химия Сергей Сергеев Аудитория 1	История Алексей Алексеев Аудитория 1	Физика Петр Петров Аудитория 1	География Николай Николаев Аудитория 1	История Алексей Алексеев Аудитория 6
16:30-18:00	География Николай Николаев Аудитория 1	Биология Дмитрий Дмитриев Аудитория 1	Литература Андрей Андреев Аудитория 1	История Алексей Алексеев Аудитория 3	Биология Дмитрий Дмитриев Аудитория 3	Биология Дмитрий Дмитриев Аудитория 4
18:10-19:40	Литература Андрей Андреев Аудитория 1	Математика Иван Иванов Аудитория 1	Физика Петр Петров Аудитория 1	Математика Иван Иванов Аудитория 2	Литература Андрей Андреев Аудитория 3	Химия Сергей Сергеев Аудитория 4

Рис. 3 – Расписание созданное «Жадным алгоритмом»

Установить ограничения

Максимум пропусков между парами:

1

Максимальное смещение начального времени:

2

Минимальное количество пар в день:

3

Максимальное количество пар в день:

5

Сохранить ограничения

Рис. 4 – Выставленные ограничения

Пары	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота
08:00-09:30			Физика Петр Петров Аудитория 1			
09:40-11:10		Биология Дмитрий Дмитриев Аудитория 7	Физика Петр Петров Аудитория 6	Химия Сергей Сергеев Аудитория 1		Химия Сергей Сергеев Аудитория 7
11:30-13:00	Биология Дмитрий Дмитриев Аудитория 6	Химия Сергей Сергеев Аудитория 2	География Николай Николаев Аудитория 1	Литература Андрей Андреев Аудитория 6	Биология Дмитрий Дмитриев Аудитория 1	
13:10-14:40		Физика Петр Петров Аудитория 1	Биология Дмитрий Дмитриев Аудитория 7		Физика Петр Петров Аудитория 1	География Николай Николаев Аудитория 1
14:50-16:20	История Алексей Алексеев Аудитория 1			Биология Дмитрий Дмитриев Аудитория 2		География Николай Николаев Аудитория 1
16:30-18:00	Химия Сергей Сергеев Аудитория 6	Физика Петр Петров Аудитория 1	География Николай Николаев Аудитория 1	География Николай Николаев Аудитория 1	География Николай Николаев Аудитория 1	Физика Петр Петров Аудитория 5
18:10-19:40			География Николай Николаев Аудитория 2			Физика Петр Петров Аудитория 1

Рис. 5 – Итог оптимизации Генетическим алгоритмом

Формирования расписания «жадным алгоритмом» не занимает много времени, другое дело оптимизация, которая при введении ограничений выполняется в среднем до 20 минут.

Улучшить оптимизацию кода можно разными способами, например улучшение производительности отдельных функций [6], использование более эффективных данных и алгоритмов и оптимизация кода для многопоточности или использования GPU или CPU. Одно из улучшений это распараллеливание вычислений, чтобы ускорить выполнение. Например, распараллеливание функции оценки: with Pool() as pool: fitness_scores = pool.starmap(fitness, [(individual, groups, days_of_week, time_slots, min_classes_per_day, max_classes_per_day, max_gaps) for individual in population])). В фрагменте кода используется способ называемый «multiprocessing.Pool» для параллельного вычисления приспособленности каждой особи в популяции, «pool.starmap» позволяет распараллелить выполнение функции «fitness», которая оценивает каждую особь это позволяет улучшить эффективность и тем самым сократить время на оценку. В коде «fitness_scores = [fitness(individual, groups, days_of_week, time_slots, min_classes_per_day, max_classes_per_day, max_gaps) for individual in population]» «фитнес» функция вычислялась последовательно для каждой особи.

В новом коде используется вероятность мутации (mutation rate), которая

была установлена на 10% и теперь она закреплена, что ограничивает количество мутаций в каждом поколении, снижая вероятность внесения случайных изменений, которые могут ухудшить качество решений, но при этом 10% - это достаточно высокая вероятность, чтобы обеспечить достаточное количество мутаций для исследования пространства решений. Выбирается случайная группа, день, временной интервал и дисциплина, проверяется возможность назначения занятия с учетом ограничений (максимальных пропусков и максимального количества занятий в день и т.д.) с помощью функции «can_schedule». Если все условия соблюдены, создается новая запись в расписании.

Выполнено уменьшение размера выборки для кроссовера и мутации также одно из нововведений. До изменения кода родители для скрещивания выбирались случайным образом из всей популяции, что могло привести к плохим потомкам. После изменения родители выбираются из верхних 20 лучших особей, что увеличивает шансы на получение более приспособленных потомков.

Данные по тестированию системы до оптимизации кода и после можно увидеть в таблице 2. В ней показано время, за которое генетический алгоритм оптимизировал составленное расписание. Данные получены за 5 попыток.

Таблица 2 – Время попыток

Номер попытки	Время попытки до внесения улучшений	Время попытки после внесения улучшений
1	20 минут 44 секунды	11 минут 13 секунд
2	19 минут 21 секунда	9 минут 55 секунд
3	23 минуты 05 секунд	10 минут 18 секунд
4	21 минута 11 секунд	8 минут 57 секунд
5	20 минут 18 секунд	10 минут 11 секунды
Среднее время	20 минут 44 секунды	10 минут 7 секунд

Проанализировав данные тестов в таблице 2, и практические результаты можно утверждать, что в среднем время на оптимизацию сократилось в 2 раза это означает что общее улучшение производительности составило около 50%. Оптимизацию алгоритма можно считать успешной.

В процессе исследования получены данные, которые могут быть

использованы для оптимизации [7] процессов управления расписанием в образовательных учреждениях [8]. Результаты заключаются в повышении эффективности организации управления расписанием и улучшения условий для обучающихся и преподавателей, а также подчеркивают важность развития информационных технологий для совершенствования управления учебными процессами.

Библиографический список:

1. Хваржев, В. М. Цифровизация информационно образовательной среды путем интеграции онлайн сервисов // Информационные технологии как основа прогрессивных научных исследований: Сборник статей Международной научно-практической конференции, Саратов, 25 июня 2024 года. – Уфа: ООО "Аэтерна", 2024. – С. 47-50. – EDN GDGCJE.
2. Постольник, В. С. Концепция автоматизированной системы составления расписания образовательной организации // Международный журнал гуманитарных и естественных наук. – 2024. – № 5-4(92). – С. 43-48. – DOI 10.24412/2500-1000-2024-5-4-43-48. – EDN NNIXUC.
3. Цебренок, К. Н. Анализ архитектуры предприятия с использованием визуальных средств моделирования // Международный журнал гуманитарных и естественных наук. – 2021. – № 8-1(59). – С. 115-118. – DOI 10.24412/2500-1000-2021-8-1-115-118. – EDN PGQPZU.
4. Цебренок, К. Н. Чат бот для обмена информацией с клиентами // Вестник ИМСИТ. – 2021. – № 4(88). – С. 24-27. – EDN PZRLIB.
5. Хваржев, В. М. Разработка концепции личного кабинета обучающегося с использованием технологий искусственного интеллекта // Международный журнал гуманитарных и естественных наук. – 2024. – № 5-4(92). – С. 147-150. – DOI 10.24412/2500-1000-2024-5-4-147-150. – EDN AJOVJE.
6. Сердюков, Е. Д. Разработка информационной системы электронных пропусков // Вестник ИМСИТ. – 2025. – № 2(102). – С. 16-21. – EDN OPOZPF.
7. Саакян, Р. Р. Проектирование информационной системы мониторинга и Дневник науки | www.dnevniknauki.ru | СМИ Эл № ФС 77-68405 ISSN 2541-8327

распределения заказов курьерской службы на основе машинного обучения // Информационные ресурсы России. – 2020. – № 3(175). – С. 34-38. – EDN NZBUQK.

8. Зибирев, В. А. Анализ средств видеоконференцсвязи для обучения с использованием дистанционных образовательных технологий // Вестник ИМСИТ. – 2024. – № 1(97). – С. 27-31. – EDN XPSSHI

Оригинальность 81%