

УДК 004.02

АВТОМАТИЗАЦИЯ ЗАДАЧИ СОСТАВЛЕНИЯ УЧЕБНОГО РАСПИСАНИЯ

Завьялов Анатолий Михайлович¹, Новиков Андрей Владимирович²

¹ Кандидат технических наук, доцент;

ГОУ ВПО «Международный Университет природы, общества и человека «Дубна», филиал «Дмитров»;
141800, Московская обл., г. Дмитров, мкр-н Махалина, д. 15;
e-mail: amzavjalov@mail.ru.

² Студент 5-го курса (магистратура по направлению «Системный анализ и управление»);

ГОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: a_newman@mail.ru.

Статья посвящена автоматизации задачи составления расписания учебных занятий для высших учебных заведений. В ней рассмотрены существующие системы, предложен алгоритм, позволяющий составить расписание, удовлетворяющее предъявляемым требованиям, разработана общая схема работы системы, структура и логическая модель базы данных, проведено тестирование системы, выявлены зависимости времени генерации расписания от количества рассматриваемых непротиворечивых вариантов и от количества учебных групп.

Ключевые слова: расписание учебных занятий, алгоритм, штрафная функция, минимизация.

AUTOMATIZATION OF PROBLEM OF TIMETABLE MAKING

Zavjalov Anatoly Mihajlovich¹, Novikov Andrey Vladimirovich²

¹ PhD, docent;

International university of the nature, society and man «Dubna», «Dmitrov» branch;
141800, Dmitrov, Moscow reg., Mahalina microdistrict, 15;
e-mail: amzavjalov@mail.ru.

² Student of 5th course (magistracy «Systems analysis and management»);

International university of the nature, society and man «Dubna»,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: a_newman@mail.ru.

Article is devoted to the automation of a problem of drawing up of the timetable for higher educational institutions. Existing systems are considered in it, the algorithm, allowing to make the timetable meeting shown requirements is offered, the general scheme of work of system, structure and logic model of a data-base is developed, system testing is held, dependences of time of generation of the timetable on quantity of considered consistent variants and on quantity of educational groups are revealed.

Keywords: the timetable of studies, algorithm, penal function, minimisation.

Задача составления расписания занятий для учебных заведений имеет давнюю историю. Она интересовала людей, занятых формированием учебного процесса с момента появления образовательных учреждений массового характера: школ, гимназий, колледжей, институтов и университетов.

Традиционно эта задача решалась человеком без использования технических средств, который на основании определенных критериев, чертил на бумаге расписание, имеющее вид таблицы и удовлетворяющее определенным требованиям. В качестве исходных данных обычно используются: список учебных групп, курсов, факультетов, учебный план (список дисциплин и количество часов в неделю для каждой дисциплины), список преподавателей, список учебных аудиторий, список требований к организации учебных занятий. Необходимо учитывать и другие факторы, например, деление студентов на группы и подгруппы, проведение потоковых лекций, проведение занятий в специализированных аудиториях, выходные дни, максимальная нагрузка в день, пожелания преподавателей о возможности проведения лекций в определённые дни недели, приглашение для проведения некоторых лекций преподавателей из других учебных заведений.

При достаточно большом наборе исходных данных и множестве ограничивающих факторов, процесс формирования расписания продолжает оставаться довольно сложным и трудоёмким занятием. Следует заметить, что в наше время всеобщей автоматизации и компьютеризации производства и общества в целом во многих образовательных учреждениях задача составления расписания решается по старинке, т. е. вручную.

Таким образом, задача автоматизация составления учебного расписания является актуальной.

В настоящее время существуют несколько подходов к решению рассматриваемой задачи. Так в работе [1] составление расписания рассматривалось в рамках линейного целочисленного программирования. Авторами этой работы задача составления расписания была разбита на пять подзадач, которые в сетке расписания занятий отражают размещение: дня самостоятельной внеаудиторной работы студентов; занятий по физическому воспитанию или другим видам занятий, проводимых вне аудиторного фонда вуза; нагрузки каждого потока по дням недели; нагрузки преподавателей по рабочим дням с учётом видов занятий для потоков и групп; взаимосвязей «учебная дисциплина — преподаватель» по номерам пар занятий для каждого дня.

В работе [2] для решения задачи применялся алгоритм имитации отжига, заимствованный из исследований поведения атомов металла в процессе его отжига.

Задачу составления расписания можно рассматривать как задачу раскраски графа [3]. Задачей раскраски графа называют поиск минимального числа цветов, необходимых для раскраски вершин некоторого графа с использованием для каждой пары соседних вершин различных цветов. Для постановки задачи составления расписания, как задачи раскраски графа, строится граф, в котором каждая вершина представляет собой запланированное учебным планом занятие. В том случае, если между какими-то двумя вершинами возможны конфликты, например, оба занятия проводятся в одной аудитории или с одним преподавателем, то они соединяются ребром. Это эквивалентно запрету одновременного проведения этих занятий. Тогда задача составления расписания представляется как минимизация числа цветов, необходимых для раскраски графа. Применение такого подхода для решения реальных задач, по-видимому, малоэффективно. В то же время, задача раскраски графа при составлении расписаний может оказаться полезной в случае её комбинации с другими алгоритмами, как это показано в работе [4].

Алгоритм составления расписания, основанный на имитационном моделировании, описан в работах [5] и [6]. Он основан на имитации действий диспетчера при составлении расписания. Алгоритм оперирует непосредственно расписанием и учебным планом. Процесс составления расписания начинается с пустого расписания, когда все занятия находятся в списке неучтённых занятий. Затем алгоритм переходит от одного незаконченного расписания к другому, стремясь наилучшим образом расставить все занятия, включённые в список. Процесс продолжается до тех пор, пока не будет сформировано полное расписание или выполнится фиксированное количество итераций. Алгоритм состоит из трёх основных шагов. На первом шаге выбирается ещё не включённое в расписание занятие. Выбор делается на основе анализа «узких мест». В данном случае «узкими местами» являются наиболее дефицитные ресурсы: студенты, преподаватели и аудитории. В первую очередь составляются расписания для наиболее дефицитных ресурсов. Это могут быть занятия, использующие дефицитный аудиторный фонд, занятия, проводимые преподавателями, которые ставят жёсткие условия по време-

ни и месту их проведения и т. д. На втором шаге для выбранного занятия определяются все возможные варианты его размещения в расписании, удовлетворяющие всем жёстким ограничениям. Далее каждая позиция оценивается с помощью специальной целевой функции, и занятие помещается в лучшую из возможных позиций. На третьем шаге, при возникновении конфликтной ситуации (на втором шаге), «конфликтующие» занятия удаляются из расписания и помещаются обратно в список неучтённых занятий. При реализации данного алгоритма должны быть разработаны правила выбора очередного занятия из списка, определения наилучшей для него позиции в расписании и оценке получаемого расписания. Такой подход к решению задачи детально учитывает специфику расписания для конкретного вуза, но при этом значительно ограничивает возможность применения этого метода в других учебных заведениях.

В работе [7] для решения задачи составления расписания применялось логическое программирование в ограничениях с использованием интеллектуального поиска наилучшего варианта расписания.

Рассмотренные методы в своей основе используют итерационную технику улучшения результатов. В течение одной итерации они ищут решение, лучшее в окрестностях данного. Если такое решение найдено, оно становится текущим и начинается новая итерация. Это продолжается до тех пор, пока прирост целевой функции (в каждой работе она своя) не уменьшится практически до нуля или не выполнится заданное количество итераций. Такие методы ориентированы только на поиск локальных оптимумов, причём положение найденного оптимума зависит от стартовой точки. Глобальный же оптимум (оптимальное расписание) может быть найден только случайно. Для повышения вероятности нахождения глобального оптимума используется множественный эксперимент с различными начальными точками, что существенно увеличивает время поиска.

В связи с этим представляет интерес разработка алгоритмов, сохраняющих преимущества описанных методов и свободных от указанного недостатка. К таким алгоритмам относятся генетические алгоритмы, в основе которых лежат стохастические эвристические оптимизационные методы, и основная идея которых взята из теории эволюционного развития видов. Использование генетических алгоритмов для решения задачи составления расписания учебных занятий представлено в работах [1] и [8].

Для составления расписания занятий в высших учебных заведениях на данный момент существует несколько программных продуктов, например: AVTOR-2003 High School Pro, АСТРА, ХроноГраф 3.0 Мастер, «Экспресс-расписание», «Составитель расписания — 2007». Указанные системы имеют следующие возможности: ручной и автоматизированный режимы составления расписания, добавления, удаления и редактирования учебных групп, списка учебных аудиторий, списков преподавателей и их пожеланий по времени работы; редактирование учебного плана и списка учебных предметов; внесение изменений в полученное расписание; экспорт и печать полученных результатов; публикация на сайте учебного заведения. Однако указанные системы не лишены недостатков, например, в них отсутствует интерфейс преподавателя, позволяющий просматривать график его работы, не учитывается вместимость аудиторий и их специфика, отсутствует возможность объединения учебных групп в потоки для проведения потоковых занятий.

Стоит заметить, что существующие программные продукты для составления расписания занятий в средних учебных заведениях не применимы в нашем случае, т. к. расписание для высших учебных заведений имеет некоторые особенности, например, наличие потоковых лекций, разбиение групп на подгруппы и т. д.

Сформулируем требования, которым должен удовлетворять разрабатываемый программный продукт:

- возможность формирования расписания, как в ручном режиме, так и в автоматизированном режиме;
- возможность редактирования полученного в результате работы системы варианта расписания;
- возможность редактирования списка кафедр, списка курсов, списка учебных групп, списка учебных аудиторий университета (список должен учитывать вместимость аудиторий и их специфику);
- возможность объединения учебных групп в потоки для проведения потоковых занятий;
- возможность редактирования списка преподавателей университета (список должен вестись с указанием рабочей ставки преподавателей и предпочтений по рабочим дням);

- возможность использования преподавателями специального интерфейса, позволяющего локально или удаленно из Интернета или локальной сети, редактировать дни, в которые он может работать, просмотреть рабочий график после составления расписания;
- возможность редактирования учебного плана, списка предметов, количества часов, отводимых для каждого предмета в неделю;
- возможность «привязки» определенного преподавателя к определенной группе и к определенному учебному предмету;
- возможность изменения числа пар в учебный день;
- возможность генерация наиболее подходящего варианта расписания с учетом учебного плана, возможностей преподавателей;
- возможность использования студентами специального интерфейса системы (локально или удаленно из Интернета), который отражает составленное расписание целиком и позволяет использовать выборки (например, отображение расписания только для определенного курса или только для определенной группы);
- возможность сохранения сгенерированного и отредактированного варианта расписания для дальнейшей работы над ним;
- возможность печати полученного расписания целиком или выборочно;
- возможность размещения полученного расписания на сайте, как целиком, так и выборочно;
- возможность экспорта расписания в формате MS Excel.

При разработке системы необходимо учесть следующие условия:

- соответствие количества пар в неделю (каждого предмета каждого преподавателя) их количеству в учебном плане;
- невозможность проведения одним преподавателем в одно и то же время занятий в нескольких группах;
- соответствие количества пар в единицу времени (равной продолжительности одной пары) у всего университета количеству свободных аудиторий в эту же единицу времени;
- аудитория, в которой проводится занятие в определенное время, должна быть свободна и в ней может разместиться вся группа, либо несколько групп при проведении потоковых лекций;
- проведение занятий в специализированных аудиториях (например, это может быть химическая лаборатория или компьютерный класс);
- возможность проведения потоковых лекций;
- возможность проведения лекций по чётным/нечётным неделям;
- возможность проведения некоторых лекций в конкретное время в определенный день недели;
- желание преподавателя вести определенные пары в определенные интервалы времени;
- оптимальное распределение количества занятий по дням для каждой из групп;
- одинаковые занятия желательно проводить непосредственно друг за другом;
- количество «окон» между занятиями должно быть минимальным.

В работе предложен метод решения задачи, основанный на выборе лучшего варианта расписания из некоторого числа случайно сгенерированных расписаний, удовлетворяющих всем поставленным требованиям.

Для оценки качества расписания предлагается использовать штрафную функцию, которая позволяет сопоставить любому варианту расписания числовой эквивалент. Штрафная функция имеет следующий вид:

$$F(x) = \sum_{i=1}^N n_i \cdot H_i, \quad (1)$$

где x — вариант расписания, N — число штрафных коэффициентов, n_i — число i -ых штрафов в данном варианте расписания, H_i — численная величина штрафа (в баллах).

Таким образом, задача поиска оптимального расписания сводится к выбору варианта расписания, для которого значение штрафной функции будет минимальным: $F(x) \rightarrow \min$.

Опрос студентов и преподавателей показал, что целесообразно ввести следующие штрафы (пара — занятие продолжительностью 1,5 астрономических часа):

- штраф за «окно»;
- штраф за отсутствие первой пары;
- штраф за вариант: более двух одинаковых пар в день;
- штраф за вариант: одна пара в день;
- штраф за вариант: две пары в день;
- штраф за вариант: три пары в день;
- штраф за вариант: четыре пары в день;
- штраф за вариант: пять пар в день;
- штраф за вариант: более пяти пар в день.

На рис. 1 представлена блок-схема алгоритма генерации расписания. Генерация вариантов расписания происходит методом ограниченного перебора, основанного на последовательном выборе для каждой пары из списка учебного плана случайного незанятого времени проведения занятия и аудитории. Стоит отметить следующее: так как время и аудитория проведения занятия выбираются случайно, то вероятность генерации двух одинаковых вариантов расписания очень мала. Алгоритм предполагает наличие приоритета по времени проведения для некоторых занятий (например, потоковые лекции). Поэтому все пары в списке учебного плана сортируются по убыванию степени приоритета. Алгоритм предполагает также проведение ежедневной минимизации количества «пустых пар».

Общая схема работы системы представлена на рис. 2.

Система включает в себя следующие компоненты:

- интерфейс учебной части;
- интерфейс преподавателя;
- интерфейс студента;
- модуль экспортирования данных на сайт;
- модули генерации расписания (основные модули).

Сотрудник учебной части, используя логин и пароль, через специальный интерфейс вводит в систему исходные данные: список кафедр, курсов, групп, аудиторий, преподавателей и учебный план. Преподаватели, используя свой интерфейс, логин и пароль, вводят в систему дни недели, в которые они могут проводить занятия в университете.

После внесения исходных данных система запускается для генерации оптимального варианта расписания. Этот вариант можно отредактировать, распечатать, сохранить и с помощью модуля экспортирования данных разместить на сайте, где произойдёт его автоматическое обновление. Студент, используя специальный интерфейс, может просмотреть всё расписание или выборочно — для группы. Система позволяет также преподавателю просмотреть график его работы.

Система может работать на одном компьютере, в локальной сети, в сети Интернет (при соответствующей настройке сервера).

На рис. 3 представлена структура и логическая модель базы данных системы, в таблице 1 приведено описание таблиц базы данных, а в таблице 2 — описание файловой структуры системы.

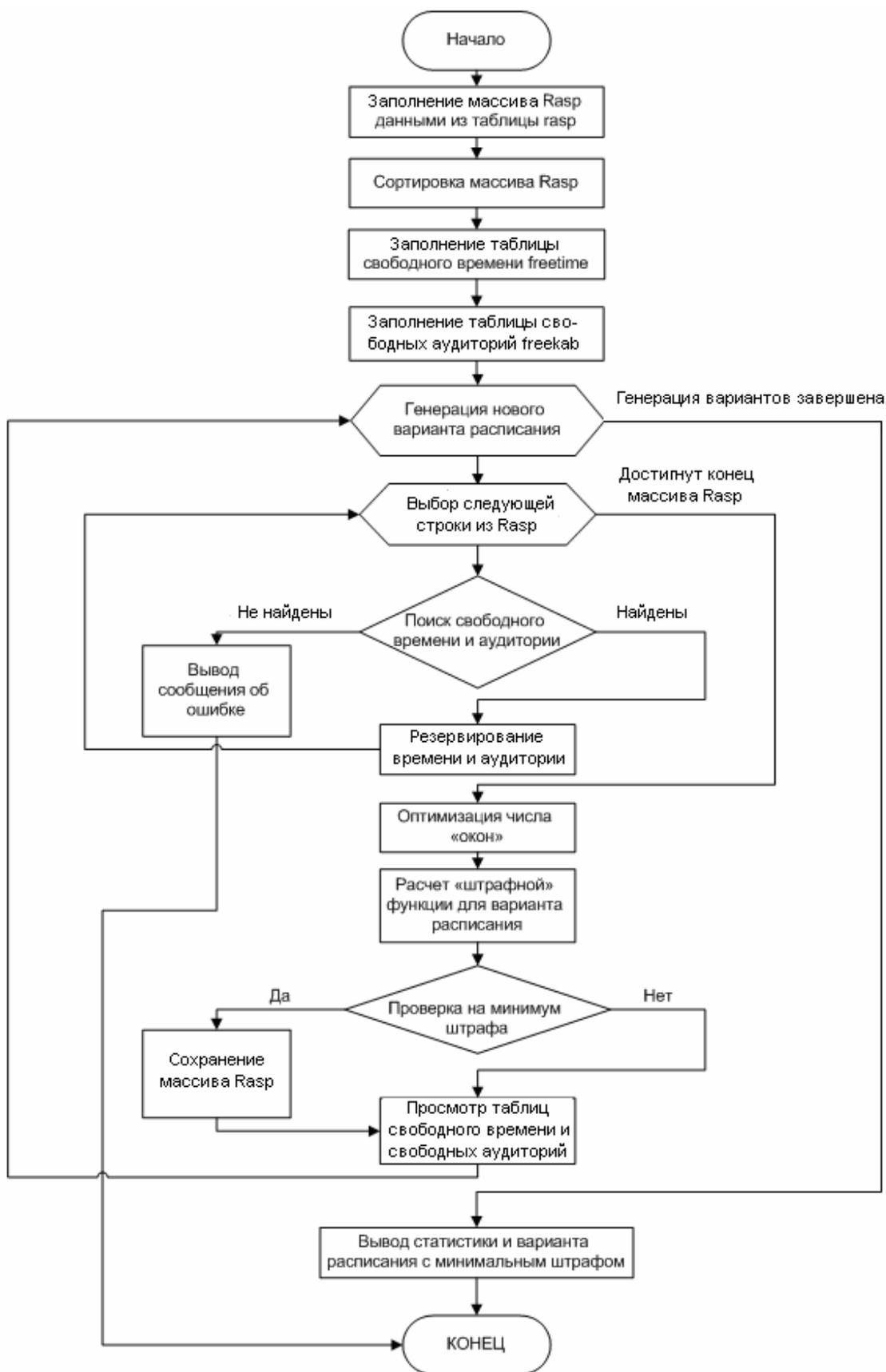
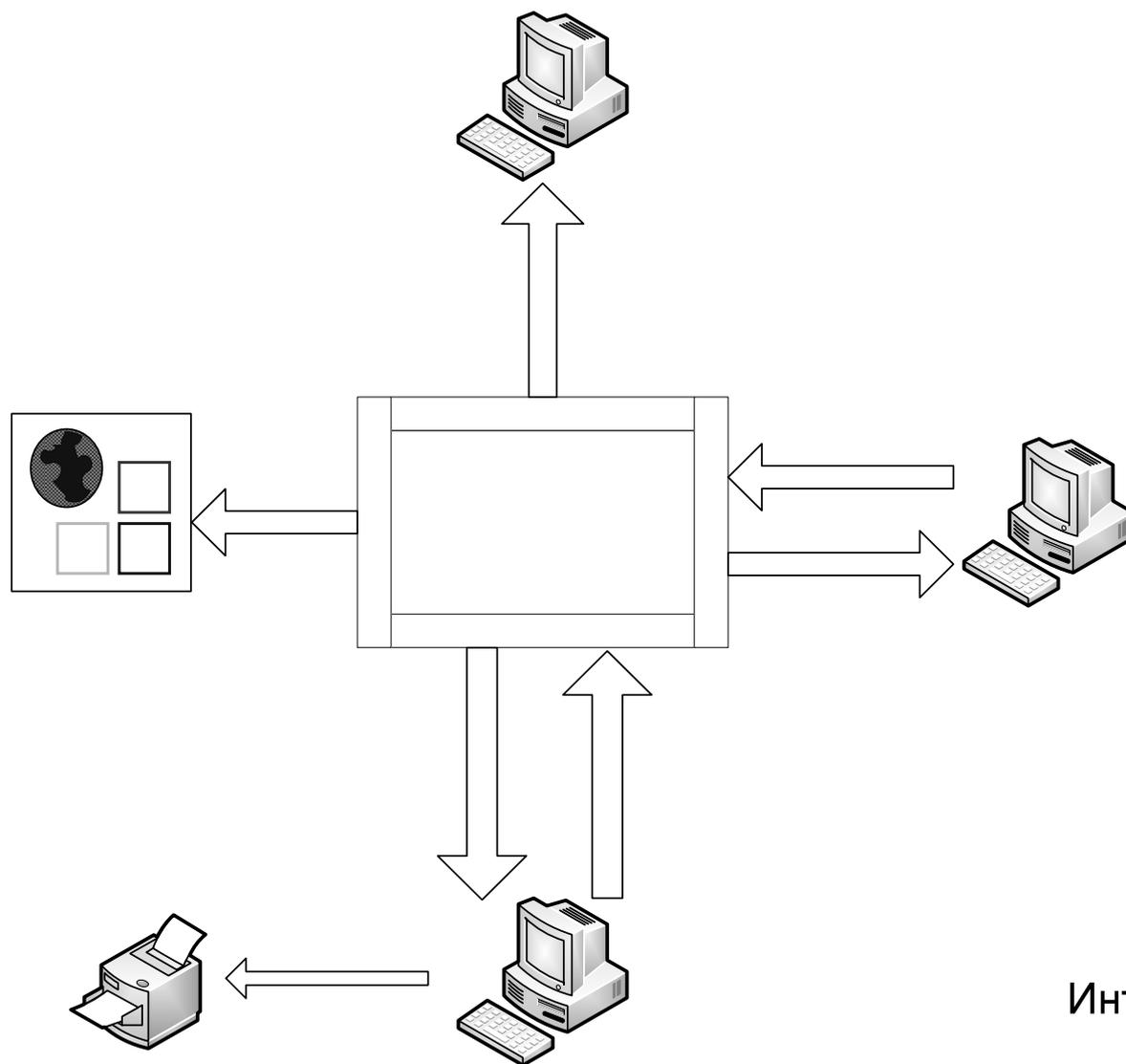


Рис. 1. Блок-схема алгоритма генерации расписания



СТУДЕНТ

Интерфейс студента

Расписание

Рис. 2. Общая схема работы системы

Данные на сайт

Основные модули системы

САЙТ УНИВЕРСИТЕТА

Интерфейс учебного расписания

Вариант расписания

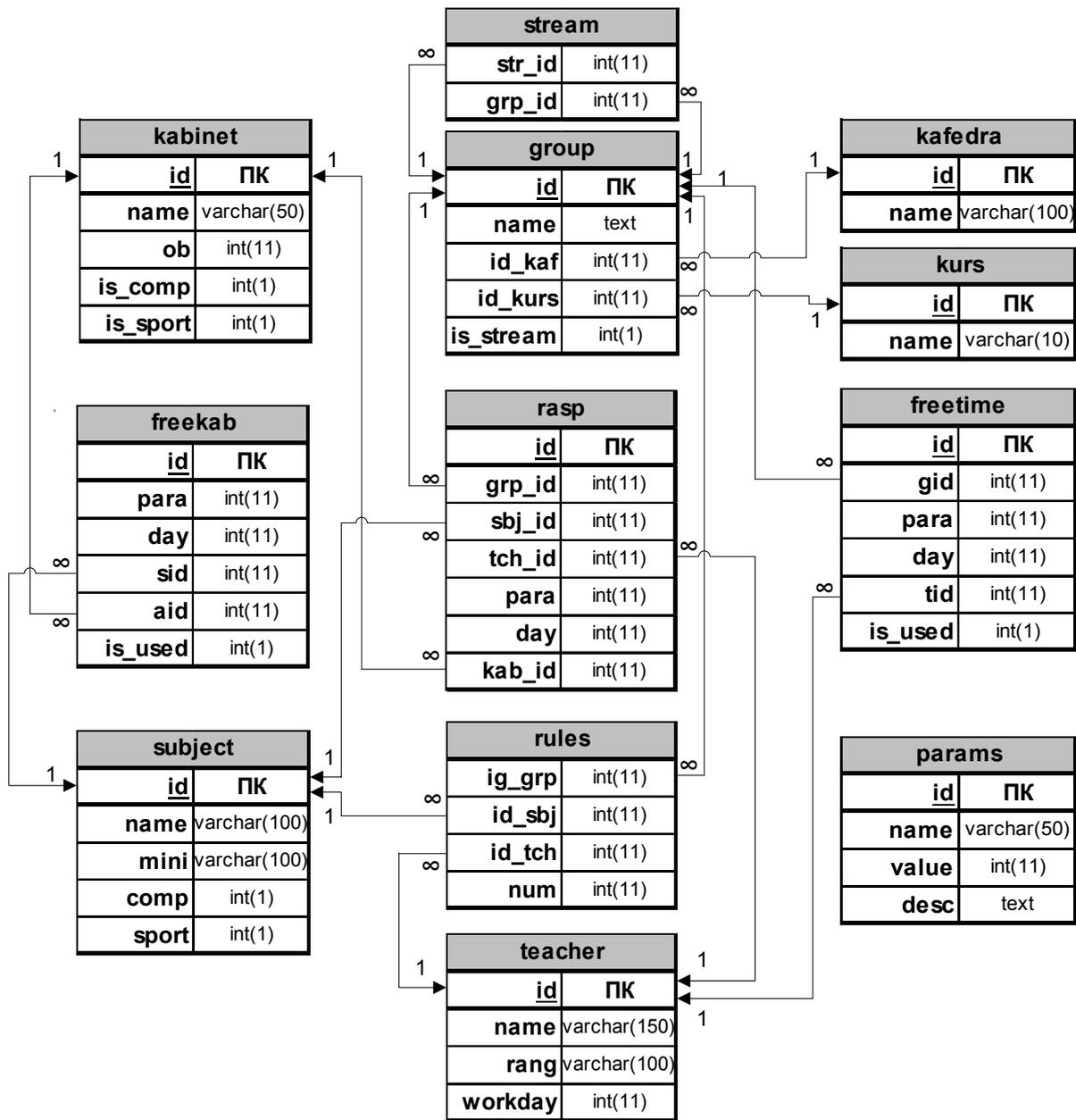


Рис. 3. Структура и логическая модель базы данных системы

Таблица 1. Описание таблиц базы данных

Таблица	Описание
freekab	Содержит все возможные комбинации связки ПАРА-ДЕНЬ-ПРЕДМЕТ-АУДИТОРИЯ. Используется для хранения промежуточных данных в процессе генерации расписания.
freetime	Содержит все возможные комбинации связки ГРУППА-ПАРА-ДЕНЬ-ПРЕПОДАВАТЕЛЬ. Используется для хранения промежуточных данных в процессе генерации расписания.
group	Содержит данные об учебных группах и описания объединенных потоков.
kabinet	Содержит данные об учебных аудиториях.
kafedra	Содержит данные о кафедрах университета.
kurs	Содержит данные об учебных курсах университета.

params	Содержит значения «штрафов» и другие параметры генерации.
rasp	Содержит сгенерированное расписание в виде набора связей ГРУППА-ПРЕДМЕТ-ПРЕПОДАВАТЕЛЬ-АУДИТОРИЯ-ПАРА-ДЕНЬ.
rules	Содержит учебный план в виде набора связей ГРУППА-ПРЕДМЕТ-ПРЕПОДАВАТЕЛЬ-КОЛИЧЕСТВО ЧАСОВ.
stream	Содержит данные об учебных потоках.
subject	Содержит данные об учебных предметах.
teacher	Содержит данные о преподавателях университета.

Таблица 2. Файловая структура системы

Имя	Описание
Includes	Директория, содержащая основные компоненты и модули системы.
connect.php	Модуль установки соединения с сервером MySQL. Содержит имя, пароль и хост для соединения, которые вводятся при установке системы.
disconnect.php	Модуль, закрывающий соединение с базой данных MySQL.
footer.php	Файл содержит HTML-код интерфейса пользователя.
gen_func.php	Модуль, содержащий основные функции для генерации расписания.
generate.php	Модуль, содержащий алгоритм генерации расписания.
group.php	Модуль управления учебными группами.
header.php	Файл содержит HTML-код верхней части интерфейса пользователя.
kabinet.php	Модуль управления учебными аудиториями.
kafedra.php	Модуль управления кафедрами университета.
kurs.php	Модуль управления учебными курсами.
menu.php	Файл содержит HTML-код пользовательского меню.
params.php	Модуль управления «штрафами» и параметрами генерации.
rules.php	Модуль управления учебным планом.
save_load.php	Модуль сохранения/загрузки сгенерированного расписания.
stream.php	Модуль управления учебными потоками.
subject.php	Модуль управления учебными предметами.
teacher.php	Модуль управления списком преподавателей.
view.php	Файл содержит HTML-код меню просмотра расписания.
I	Директория, содержащая графические элементы пользовательского интерфейса.
index.php	Основной файл – подключает все модули системы.
excel.php	Скрипт экспортирования расписания в формат MS Excel.
print.php	Скрипт, выводящий расписание на экран или на принтер.
Tmp	Директория, содержащая временные файлы.

Следует отметить, что в процессе генерации вариантов расписаний активно используются выборки из таблиц посредством оператора SELECT, поэтому для увеличения быстродействия процесса генерации использовалось индексирование полей. Эта технология позволяет на порядок уменьшить время изъятия выборки из таблицы за счет хранения индексов записей непосредственно на жестком диске компьютера.

Разработанная система работает следующим образом. После запуска в окно браузера загружается меню системы и форма, позволяющая осуществлять такие функции, как добавление, удаление и редактирование сведений о кафедрах (рис. 4).

№	Название кафедры	Команда
1	Государственное и муниципальное управление	
2	Математика и прикладная информатика	
3	Менеджмент	

Добавить

Рис. 4. Форма «Кафедры университета»

Пункт меню «Курсы» позволяет открыть форму для работы со списком курсов университета: пользователь может добавлять записи в список, удалять их и редактировать список. Форма «Курсы университета» представлена на рис. 5.

После формирования списков курсов и кафедр в систему вводятся сведения об учебных группах. Добавление новой группы происходит с указанием курса и кафедры. Внесенные данные могут быть отредактированы. Если количество групп больше десяти, то для удобства пользователя список разбивается на страницы. Форма для работы со списком групп представлена на рис. 6.



Рис. 5. Форма «Курсы университета»



Рис. 6. Форма «Учебные группы университета»

Если учебный план предусматривает проведение потоковых лекций, то пользователь обязан указать группы, которые система должна объединить. Это можно сделать с помощью формы, представленной на рис. 7. Здесь можно добавлять, удалять и редактировать записи списка объединения групп в потоки. При удалении какой-либо группы из списка групп, она автоматически удаляется из списка всех потоков.

№	Состав	Команда
1	531, 532	
2	131, 132, 133	
3	231, 232	

Добавить

Рис. 7. Форма «Объединения групп для потоковых занятий»

Пункт меню «Аудитории» позволяет открыть форму для работы со списком аудиторий университета (рис. 8). Здесь отображается вместимость аудиторий и их специфика, т. е. является ли она компьютерной аудиторией или спортивным залом. Пользователь может добавлять записи в список, удалять их и редактировать список.

Выполнение пункта меню «Предметы» позволяет открыть форму «Учебные предметы университета» (рис. 9). Здесь для каждого предмета указывается название, сокращенное название, которое используется для отображения в расписании, необходимость проведения занятия в компьютерной аудитории или в спортзале.

Учебные аудитории университета:

№	Название	Вместимость	Комп. кабинет	Спортзал	Команда
1	205	40	НЕТ	НЕТ	
2	206	40	НЕТ	НЕТ	
3	207	40	НЕТ	НЕТ	
4	209	40	НЕТ	НЕТ	
5	212	15	ДА	НЕТ	
6	213	15	ДА	НЕТ	
7	215	40	НЕТ	НЕТ	
8	216	40	НЕТ	НЕТ	
9	218	60	НЕТ	НЕТ	
10	спортзал	60	НЕТ	ДА	

Добавить

Рис. 8. Форма «Учебные аудитории университета»

Учебные предметы университета:

№	Название	Сокращенное название	Нужен комп. кабинет	Нужен спортзал	Команда
1	Алгебра и геометрия - лекция	АЛГ.ГЕОМ.(л)	НЕТ	НЕТ	
2	Алгебра и геометрия - семинар	АЛГ.ГЕОМ.(с)	НЕТ	НЕТ	
3	Английский язык	АНГЛ.ЯЗ.	НЕТ	НЕТ	
4	Базовые технологии программирования -	БТП	ДА	НЕТ	
5	Базы данных - лекция	БД(л)	НЕТ	НЕТ	
6	Базы данных - семинар	БД(с)	ДА	НЕТ	
7	Безопасность жизнедеятельности	БЖД	НЕТ	НЕТ	
8	Биология - лекция	БИОЛОГИЯ(л)	НЕТ	НЕТ	
9	Биология - семинар	БИОЛОГИЯ(с)	НЕТ	НЕТ	
10	Вычислительная	ВЫЧ.МАТ.(л)	НЕТ	НЕТ	

Рис. 9. Форма «Учебные предметы университета»

Форма «Преподаватели университета» (рис. 10) отображает сведения о преподавателях, их пожелания по проведению занятий в определённые дни недели. Пожелания учитываются при генерации расписания.

Преподаватели университета:					
№	ФИО	Должность	Рабочие дни	Команда	
1	Антонов Владимир Иванович	Старший преподаватель	Понедельник, Среда, Суббота		
2	Банаева Наталья Ивановна	старший преподаватель	Среда, Четверг		
3	Брискин Владимир Львович	доцент	Вторник, Среда, Четверг		
4	Гребенкин Михаил Федорович	Профессор	Понедельник, Вторник		
5	Завьялов Анатолий Михайлович	доцент	Вторник, Среда, Пятница		
6	Зенченко Светлана Васильевна	старший преподаватель	Среда, Пятница, Суббота		
7	Зубанов Алексей	Стажер	Среда		

Рис. 10. Форма «Преподаватели университета»

Форма «Учебный план» предназначена для ввода в систему учебного плана (рис. 11). Она содержит таблицу с полями «Группа», «Предмет», «Преподаватель», «Количество пар». Таблица заполняется построчно: для каждой группы вводятся данные по предметам, преподавателям, количеству пар в неделю. В поле «Группа» вводятся также данные для потоков: указываются предметы, преподаватели и количества пар. В таблицу можно добавлять строки, а данные в них — редактировать. Каждое изменение должно быть зафиксировано нажатием на кнопку «Сохранить».

Учебный план					
	Группа	Предмет	Преподаватель	Кол-во пар	
гели	431	Физкультура	Антонов Владимир Иванович	1	Удалить
	431	Сети ЭВМ и телекоммуникации - лек	Митрошин Павел Алексеевич	1	Удалить
	431	Сети ЭВМ и телекоммуникации - сем	Митрошин Павел Алексеевич	1	Удалить
лан	431	Методы и средства защиты информ.	Щетинина Дарья Петровна	1	Удалить
	431	Теория систем и системный анализ	Щетинина Дарья Петровна	1	Удалить
	431	Теория систем и системный анализ	Щетинина Дарья Петровна	1	Удалить
	431	Английский язык	Осадченко Валентин Гаврилович	1	Удалить
	431	Системы искусственного интеллект.	Мещериков В.А.	1	Удалить
	431	Системы искусственного интеллект.	Мещериков В.А.	1	Удалить
	431	Базовые технологии программирова	Марков Кирилл Николаевич	1	Удалить
	431	Программирование в UNIX - лекция	Марков Кирилл Николаевич	1	Удалить
	431	Программирование в UNIX - семинар	Марков Кирилл Николаевич	2	Удалить
	531, 532	Вычислительная математика - лекци	Ширков Петр Дмитриевич	1	Удалить

Рис. 11. Форма «Учебный план»

Параметры генерации расписания устанавливаются на форме «Редактирование параметров генерации расписания» (рис. 12). Здесь указываются значения штрафов, количество генерируемых вариантов, максимальное количество пар в день и параметр, отвечающий за проведение занятий в компьютерных аудиториях. Чтобы зафиксировать введенные параметры, необходимо нажать на кнопку «Сохранить».

После ввода параметров генерации, можно приступить непосредственно к генерации расписания, выполнив пункт меню «Генератор». Форма с результатами генерации представлена на рис. 13. Оптимальный вариант расписания (с точки зрения минимума штрафной функции) можно просмотреть (рис. 14), записать во временный файл, восстановить из временного файла, сохранить в формате csv для редактирования в табличном процессоре MS Excel. Процесс генерации можно повторить.

Составление расписания

Редактирование параметров генерации расписания

Описание параметра	Значение
Штраф за "окно":	10000
Штраф за первую пустую пару:	60
Штраф за одинаковые пары:	10
Штраф за одну пару в день:	200
Штраф за две пары в день:	60
Штраф за три пары в день:	10
Штраф за четыре пары в день:	100
Штраф за пять пар в день:	300
Штраф за кол-во пар больше пяти:	1000
Кол-во генерируемых вариантов:	10
Максимальное кол-во пар в день:	6
Разрешить проводить занятия, не требующие компьютера:	

Рис. 12. Форма «Редактирование параметров генерации расписания»

Составление расписания

Результат генерации

Сгенерировано 100 вариантов расписания

Найден вариант расписания с минимальным штрафным баллом, равным 15790

Максимальное кол-во пар в день - 6
 Время генерации - 148.46734786 сек.

Просмотр расписания

Сохранить результат во временном файле
 Запустить генерацию снова

Восстановить результат из временного файла

Рис. 13. Форма «Результат генерации расписания»

	131	132	133	231	232	531
П о н е д	ПЯВУ(с) - 213	ОЭВМ(с) - 212		АНГЛ.ЯЗ. - 205	КТАДС(с) - 216	
	ПЯВУ(п) - 206	ПЯВУ(п) - 206	ПЯВУ(п) - 206	БД(п) - 215	БД(п) - 215	КТД(с) - 213
	АНГЛ.ЯЗ. - 206				ФИЗ-РА - спортзал	
						ИТ(п) - 216
В т о р	ЭКОНОМ.(с) - 218	ЭКОНОМ.(с) - 218	ЭКОНОМ.(с) - 218	САОД(п) - 216	САОД(п) - 216	ВЫЧ.МАТ.(с) - 212
	ИНФОРМ.(п) - 209	ИНФОРМ.(п) - 209	ИНФОРМ.(п) - 209	ФИЗИКА(с) - 206	БД(с) - 213	ОП(п) - 218
	ФИЗ-РА - спортзал		АНГЛ.ЯЗ. - 216		ТПР(с) - 212	БЖД - 209
	БИОЛОГИЯ(с) - 206	БИОЛОГИЯ(с) - 206	БИОЛОГИЯ(с) - 206	ТПР(п) - 207	ТПР(п) - 207	
	ОИТ(с) - 212			АНГЛ.ЯЗ. - 209		
С р е д а	ОЭВМ(п) - 218	ОЭВМ(п) - 218	ОЭВМ(п) - 218	ДМ(п) - 206	ДМ(п) - 206	ОМнМ - 205
	АНГЛ.ЯЗ. - 218	ПЯВУ(с) - 212	АНГЛ.ЯЗ. - 216			ЭЛ.ТЕХ.(с) - 206
	АЛГ.ГЕОМ.(с) - 207	АНГЛ.ЯЗ. - 209	АЛГ.ГЕОМ.(с) - 218	КТАДС(с) - 216	ФИЗ-РА - спортзал	ПТИ(с) - 213
	ФИЗ-РА - спортзал	ОИТ(с) - 212		ФИЗИКА(с) - 207	ДМ(с) - 216	
			МАТАН(с) - 207	ФИЗ-РА - спортзал		
Ч е т	МАТАН(с) - 209		ОЭВМ(с) - 212	КТАДС(п) - 205	КТАДС(п) - 205	МС(с) - 213
	ОЭВМ(с) - 213	АНГЛ.ЯЗ. - 205	ФИЗ-РА - спортзал		АНГЛ.ЯЗ. - 218	ВЫЧ.МАТ.(п) - 209
	АЛГ.ГЕОМ.(п) - 216	АЛГ.ГЕОМ.(п) - 216	АЛГ.ГЕОМ.(п) - 216		САОД(с) - 212	ОП(с) - 213
		АНГЛ.ЯЗ. - 206		ДМ(с) - 216	АНГЛ.ЯЗ. - 218	
		ОИТ(с) - 213	ФИЗ-РА - спортзал		МС(п) - 218	
	ФИЗ-РА - спортзал		ФИЗИКА(с) - 216	ФИЗИКА(с) - 216	КТАДС(с) - 206	

Рис. 14. Фрагмент сгенерированного расписания

Алгоритм генерации расписания позволяет выявить ошибки, при которых сгенерировать расписание невозможно. При появлении таких ошибок рекомендуется запустить процесс генерации еще раз, а при последующем их появлении пересмотреть учебный план, рабочие дни преподавателей, список учебных аудиторий и списки групп.

При тестировании в систему были введены данные для генерации расписания занятий восьми учебных групп студентов (направление — ИВТ, второй семестр 2007-2008 учебного года). Использовались следующие входные данные:

- список преподавателей — 23 преподавателя;
- список предметов — 68 предметов (лекция и семинар - разные предметы);
- список групп — 8 групп, которые включают 3 потока по 2, 2 и 3 группы;
- список аудиторий — 10 аудиторий, которые включают две компьютерные аудитории и спортзал;
- учебный план.

Генерация проводилась на компьютере со следующими характеристиками и программным обеспечением:

- процессор — Intel Core 2 Duo T5500 1,66 GHz;
- объем оперативной памяти — 1536 Mb;
- операционная система — MS Windows XP Service Pack 2;
- http-сервер Apache v. 2.2.4, PHP v. 5.2.4, MySQL сервер v.5.0.45.
- значения параметров генерации приведены в таблице 4.

Таблица 4. Параметры тестовой генерации расписания

Параметр	Значение параметра, баллы
Штраф за окно:	1000
Штраф за первую пустую пару:	60
Штраф за одинаковые пары:	10
Штраф за одну пару в день:	300
Штраф за две пары в день:	60
Штраф за три пары в день:	10
Штраф за четыре пары в день:	60
Штраф за пять пар в день:	800
Штраф за количество пар больше пяти:	1000
Количество генерируемых вариантов:	1000
Максимальное количество пар в день:	6

Для преподавателей в системе была установлена 6-дневная рабочая неделя, проведение обычных занятий в компьютерных аудиториях было запрещено.

Проведя генерацию 1000 вариантов расписаний, было получено оптимальное расписание со значением штрафной функции равным 11640 баллам и 7 «окнами». Время генерации составило примерно 1473 секунды.

Расписание, составленное методистом учебной части, содержало 10 «окон», значение штрафной функции для него составило 19600 баллов.

Стоит заметить, что после автоматической генерации, полученное расписание можно отредактировать, тем самым еще больше минимизировав значение штрафной функции и, следовательно, получить наиболее оптимальный вариант расписания.

В работе проводилось исследование зависимости между количеством генерируемых вариантов расписания и общим временем генерации. Для этого автоматически измерялось время генерации расписания для 8 учебных групп при 10, 50, 100, 500 и 1000 вариантах.

Результаты, приведённые на рис. 15, показывают, что зависимость между временем генерации и количеством рассматриваемых вариантов линейная, то есть, при увеличении количества рассматриваемых вариантов, время поиска оптимального расписания будет увеличиваться линейно. Это позволяет проводить поиск при довольно большом количестве генерируемых вариантов и получать наиболее оптимальный вариант расписания.

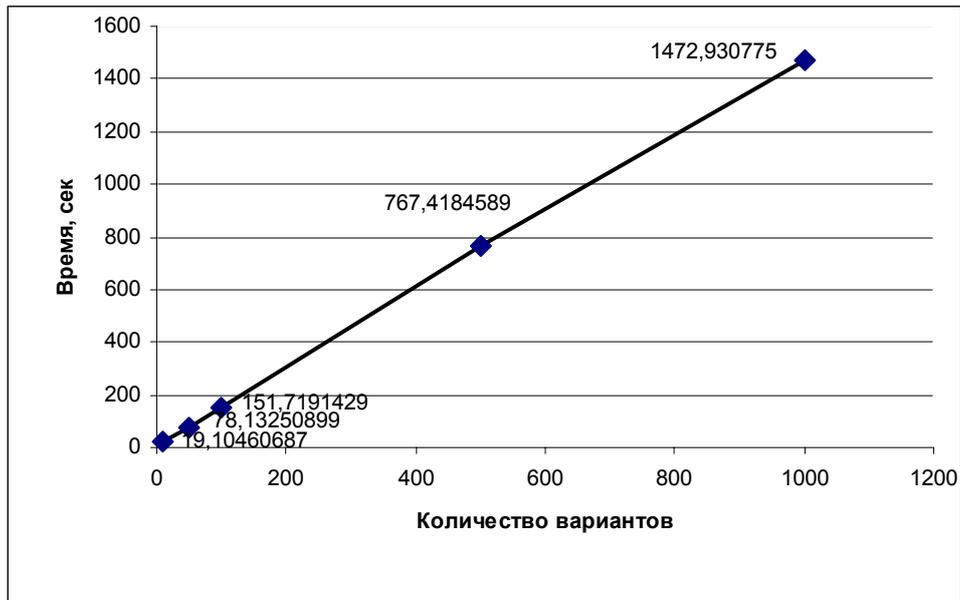


Рис. 15. График зависимости времени генерации расписания от количества рассматриваемых вариантов

Для исследования зависимости между количеством учебных групп и временем генерации, проводилась генерация 500 вариантов расписания для 3, 5, 7 и 8 групп. Полученный результат представлен на рис. 16. Зависимость носит слабый экспоненциальный характер (показатель роста равен 0,2457) и близка к линейной зависимости.



Рис. 16. Зависимость между количеством учебных групп и временем генерации расписания

Таким образом, создана система, позволяющая в автоматизированном режиме составлять расписания учебных занятий для высших учебных заведений. В работе рассмотрены существующие системы, предложен алгоритм, позволяющий составить расписание, удовлетворяющее предъявляемым требованиям, разработана общая схема работы системы, структура и логическая модель базы данных, проведено тестирование системы, выявлены зависимости времени генерации расписания от количества рассматриваемых непротиворечивых вариантов и от количества учебных групп.

Список литературы

1. Балтак, С. В. Построение расписаний учебных занятий на основе раскраски вершин графа [Текст] / С. В. Балтак, Ю. Н. Сотсков // Информатика. Объединенный институт проблем информатики НАН Беларуси. — 2006. — № 3(11).
2. Логоша, Б. А. Комплекс моделей и методов оптимизации расписания занятий в вузе [Текст] / Б. А. Логоша, А. В. Петропаловская // Экономика и математические методы. — 1993. — Т.29, №4.
3. Низамова, Г. Ф. Математическое и программное обеспечение составления расписания учебных занятий на основе агрегативных генетических алгоритмов: автореф. дис. ... канд. тех. наук / Г. Ф. Низамова; Уфим. гос. авиац.-техн. ун.-т. — Уфа, 2006.
4. Рубальская, О. Н. Автоматизированные системы составления учебных расписаний [Текст] / О. Н. Рубальская. — М. : Наука, 2001.
5. Уоссермен, Ф. Нейрокомпьютерная техника. Теория и практика [Текст] / Ф. Уоссермен. — М. : Мир, 1992.
6. Burke, E. K. A University Timetabling System Based on Graph Colouring and Constraint Manipulation / E. K. Burke, D. G. Elliman, R. F. Weare // Journal of Research on Computing in Education, 1993.
7. Muller, T. Some Novel Approaches to Lecture Timetabling / T. Muller // In Proceedings of the 4th Workshop of Constraint Programming for Decision and Control, Gliwice, September, 2002.
8. Stamatopouls, P. Nearly Optimum Timetable Construction Through CLP and Intelligent Search / P. Stamatopouls, E. Viglas, S. Karaboyas // International Journal on Artificial Intelligence Tools. Vol. 7. 1998. — №. 4.