

УДК 004.021

doi:10.18720/SPBPU/2/id24-486

Жабко Георгий Петрович¹,
старший преподаватель;
Сиднев Александр Георгиевич²,
доцент, канд. техн. наук, доцент;
Хомяков Виктор Сергеевич³,
магистрант

АВТОМАТИЗАЦИЯ СОСТАВЛЕНИЯ РАСПИСАНИЯ В ВУЗЕ С ИСПОЛЬЗОВАНИЕМ SMT-РЕШАТЕЛЕЙ И ГЕНЕТИЧЕСКОГО АЛГОРИТМА

^{1,2,3} Россия, Санкт-Петербург,

Санкт-Петербургский политехнический университет Петра Великого,

¹ g_zhabko@mail.ru, ² sidnev_ag@spbstu.ru, ³ homyakov.vs@edu.spbstu.ru

Аннотация. Работа посвящена формированию математической модели учебного расписания высшей школы или иного аналогичного подразделения вуза, а также разработке и исследованию алгоритмов составления расписания на ее основе. Этапы разработки рассматриваемой темы:

1) анализ существующих моделей и алгоритмов составления расписания учебных занятий в вузе;

2) разработка математической модели расписания вуза в рамках Высшей школы компьютерных технологий и информационных систем (ВШ КТиИС);

3) разработка и исследование алгоритмов составления учебного расписания на основе предложенной модели.

В результате представлена математическая модель расписания, совокупность объектов и ограничений, являющаяся отражением учебного процесса в вузе. Исследованы существующие подходы к построению расписания, реализованы собственные методы решения задачи на основе генетического алгоритма и использования SMT-решателей.

Ключевые слова: теория расписаний, математическая модель расписания вуза, задача составления расписания вуза, автоматизация составления расписания, генетический алгоритм, SMT-решатель.

George P. Zhabko¹,
Senior Lecturer;
Alexander G. Sidnev²,
Associate Professor, Candidate of Technical Sciences;
Victor S. Khomyakov³,
Master's Student

AUTOMATION OF UNIVERSITY SCHEDULE DEVELOPMENT USING SMT SOLVERS AND GENETIC ALGORITHM

^{1,2,3} Peter the Great St.Petersburg Polytechnic University, St. Petersburg, Russia,
¹ g_zhabko@mail.ru, ² sidnev_ag@spbstu.ru, ³ homyakov.vs@edu.spbstu.ru

Abstract. The work is devoted to the formation of a mathematical model of the academic schedule of a higher school or other similar department of a university, as well as the development and research of scheduling algorithms based on it. Stages of the development of the topic under consideration:

- 1) research of existing university course scheduling model and algorithms;
- 2) development of the mathematical model within the confines of university higher school;
- 3) development of the university courses scheduling algorithms according to described model.

As a result, a mathematical model of the schedule, a set of objects and restrictions, which is a reflection of the educational process at the university, is presented. The existing approaches to scheduling are investigated, and proprietary methods for solving the problem based on a genetic algorithm and the use of SMT solvers are implemented.

Keywords: scheduling theory, mathematical model for university course scheduling, university course scheduling problem, scheduling automatization, genetic algorithm, SMT solver.

Введение

Актуальность задачи автоматизации составления расписания занятий определяется тем, что повсеместно растут требования к качеству учебного процесса, которое зависит от эффективного распределения и планирования работы студентов и преподавателей. На сегодняшний день учебное расписание в вузах часто составляется диспетчерами учебного отдела вручную. Этот процесс занимает существенное количество времени и требует вмешательства квалифицированных специалистов.

Представленные на рынке продукты, позволяющие автоматизировать составление расписания, имеют свои сильные стороны и могут значительно облегчить работу диспетчера учебного отдела. Однако, в каждом вузе присутствует собственная специфика учебного процесса, сложные уникальные требования, не все из которых могут учитываться данными программами при составлении расписания.

Обзор средств автоматизации составления расписания

При составлении расписания в высшей школе используется система «Галактика РУЗ» [2], которая обладает возможностями автоматического составления расписания. Однако из-за сложности представления исходных данных об учебном процессе в необходимом формате и иных причин эти возможности не используются и составление расписания в рамках высшей школы происходит в ручном режиме.

На рынке присутствуют и другие аналогичные программные средства. Среди наиболее популярных решений можно выделить «1С: Автоматизированное составление расписания. Университет» [1], «Экспресс-расписание ВУЗ» [8], «AVTOR High School» [5].

Данные программные решения имеют свои сильные стороны и могут существенно упростить работу по составлению расписания. Однако, ни одно из представленных решений не является универсальным: каждый вуз, кафедра и высшая школа имеют свои особенности, которые влияют на образовательный процесс и составление расписания.

Задача составления расписания

На рубеже XX и XXI веков актуальным стало создание систем автоматизированного управления учебным процессом в образовательных системах массового обучения [7]. Связано это с усилением требований к качеству обучения, появлением разнообразных форм обучения, развитием форм дистанционного обучения, необходимостью повышения экономической эффективности обучения и др.

Составление расписания представляет собой задачу целочисленного программирования, сложность которой увеличивается экспоненциально при росте числа возможных значений переменных (эта задача относится к классу NP-сложных). Кроме того, она характеризуется большим объёмом разнородной исходной информации и множеством трудно формализуемых требований. Эти факторы затрудняют автоматизацию процесса составления расписания, несмотря на наличие множества методов целочисленного программирования [6].

Формулирование требований к модели расписания

Каждый вуз, так же, как и каждая кафедра, и высшая школа в отдельности имеют свою специфику проведения занятий и построения расписания. Для определения ключевых особенностей расписания учебных занятий, которые должны быть отражены в математической модели, предлагается следующий список требований:

- 1) произвольный период проведения занятий, не привязанный к сетке чётных и нечётных недель;
- 2) возможность проведения занятий в любой части семестра;

3) возможность проведения «блочных» занятий, требующих несколько подряд идущих пар;

4) учёт возможного разбиения занятий на подгруппы, а также объединения в потоки;

5) учёт типов и количества различного оборудования, требуемого для проведения занятий.

Математическая модель расписания

Модель расписания занятий представляет собой совокупность взаимосвязанных объектов и ограничений, описывающих учебный процесс в вузе.

Объекты представляют собой сущности, на взаимодействии которых строится учебный процесс. Для задачи составления расписания вуза в качестве основных сущностей можно выделить преподавателей, группы студентов, аудитории и занятия. Очевидно, что не любая композиция объектов и их связей может считаться корректным расписанием занятий. За это отвечает вторая часть модели – ограничения.

Ограничения характеризуют связи между объектами и критерии, которым эти связи должны удовлетворять. При соблюдении ограничений гарантируется получение корректного расписания учебного процесса.

Структура разработанной модели и ограничений во многом основана на работе Ю. С. Кабальнова [3]. Развивая идею автора, модель была существенно изменена и дополнена как новыми объектами, так и ограничениями.

Объекты математической модели

Модель оперирует пятью типовыми математическими объектами:

1) цикл занятий: включает в себя информацию об участниках учебного процесса (преподавателях и группах студентов), дисциплине, длительности и частоте занятий, объединения занятий в потоки и цепочки;

2) временной интервал: содержит информацию о конкретной паре учебного семестра (номер недели, дня и пары в учебном дне);

3) аудитория: определяет номер учебного корпуса, номер аудитории и её вместимость;

4) оборудование аудитории: включает номер соответствующей аудитории, тип и количество оборудования;

5) требование к оборудованию: сведения о количестве и типах оборудования, необходимого для проведения определенного занятия.

Объекты модели расписания по типам организуются в множества. Конкретные характеристики объекта описываются его компонентами. Пример структуры объекта типа «Аудитория»:

$$A(\text{Auditory}) = \{a^j\}, \quad a^j = (a_b^i, a_n^i, a_c^i), \quad j = \overline{1, N_a}, \quad (1)$$

где компоненты объекта: α_b^j (building) – номер учебного корпуса; α_n^j (auditory number) – номер аудитории в корпусе; α_c^j (capacity) – вместимость аудитории.

Кодирование вектора расписания

С учетом приведенных типов объектов, вводятся два целочисленных вектора α и τ , которыми полностью определяется расписание занятий:

$$\alpha = (\alpha_1, \alpha_2, \dots, \alpha_i, \dots, \alpha_{N_c}), \quad \tau = (\tau_1, \tau_2, \dots, \tau_i, \dots, \tau_{N_c}) \quad (2)$$

где $\alpha_i \in A$ – код аудитории, назначенной i -му циклу занятий, а $\tau_i \in T$ – код пары, назначенный первому занятию из i -го цикла занятий.

Жёсткие ограничения модели

Жёсткие ограничения модели есть требования, нарушение которых приводит к невозможности проведения занятий. К жёстким ограничениям относятся:

- 1) отсутствие накладок для учебных групп,
- 2) отсутствие накладок для аудиторий,
- 3) соответствие аудитории множеству допустимых аудиторий,
- 4) соответствие вместимости аудитории размеру групп,
- 5) наличие необходимого оборудования в аудитории,
- 6) отсутствие накладок для преподавателей,
- 7) необходимость проведения занятий в полном объеме,
- 8) необходимость проведения занятий в заданной части семестра.

Жёсткие требования представлены в виде предикатов относительно компонентов векторов α и τ . Далее приводится пример записи жёсткого ограничения на отсутствие накладок для аудиторий.

Пример. Для любого временного интервала t^k и любой аудитории a^r существует либо единственное занятие, которое проходит в этой аудитории в это время, либо в этой аудитории в это время не проходит ни одного занятия, либо все занятия, проходящие в этой аудитории в это время, являются элементами одного поточного занятия (с совпадающим кодом потока).

$$\begin{aligned} & \forall (a^r, t^k): a^r \in A, t^k \in T \\ & \left(\exists! c^i: (\alpha^i = a^r) \wedge (c^i \in C^{t^k}) \right) \vee \left(\neg \exists c^i: (\alpha^i = a^r) \wedge (c^i \in C^{t^k}) \right) \\ & \quad (\forall (c^i, c^j): (\alpha^i = \alpha^j = a^r) c_{cc}^i = c_{cc}^j) \end{aligned} \quad (3)$$

Мягкие ограничения

Мягкие ограничения модели суть требования, выполнение которых не является обязательным, но позволяет оптимизировать учебный процесс. Мягкие ограничения модели в общем виде вводятся как некоторые функции оценки расписания.

Оптимизация учебного расписания с использованием мягких ограничений предполагает нахождение экстремума взвешенной суммы функций оценки.

Очевидным недостатком такого определения композитного критерия качества расписания является сложность настройки весов. Заранее невозможно однозначно определить, какие из критериев являются наиболее и наименее важными, как должны соотноситься веса между разными критериями для получения расписания, соответствующего качественному учебному процессу с точки зрения мягких ограничений.

В виду того, что подобная проблема не имеет прямого решения, для получения некоторых субоптимальных значений весов предлагается разработать и использовать интерактивную процедуру, которая могла бы калибровать веса критериев как абсолютно, так и друг относительно друга, в процессе диалога с диспетчером. Этот процесс также можно частично автоматизировать, используя в качестве ответов на вопросы информацию о расписаниях прошлых лет.

Среди мягких ограничений модели можно выделить следующие:

- 1) соответствие пожеланиям преподавателей,
- 2) минимизация числа «окон»,
- 3) равномерность занятий в течение недели,
- 4) минимизация перемещений между учебными корпусами.

Существующие алгоритмы составления расписания

Решения задач, связанных с составлением расписания выполнения работ в производственных системах, привело к созданию теории решения задач составления расписаний (ЗСР) [7]. Решение ЗСР в рамках данной теории в итоге сводится к использованию математического аппарата целочисленного программирования [4, 9].

Интенсивно развиваются в последнее время методы решения больше-размерных задач целочисленного программирования, объединённых термином «генетические алгоритмы» [7]. Основные отличия и преимущества генетических алгоритмов заключаются в следующем:

- 1) генетический алгоритм (ГА) работает с кодами, в которых представлен набор параметров, напрямую зависящих от аргументов целевой функции;
- 2) в процессе поиска ГА использует несколько точек поискового пространства (процесс распараллеливается), а не переходит от точки к точке, как это происходит в традиционных методах;
- 3) ГА в процессе работы не использует дополнительной информации, что повышает скорость его работы.

Разработка алгоритма составления расписания

Получение изначально хорошего с точки зрения мягких критериев расписания, удовлетворяющего всем жестким ограничениям – неподъемная задача, поскольку пространство решений слишком велико, малая часть которых удовлетворяет всем необходимым требованиям. По этой причине принято решение о разделении процесса построения расписания на две фазы: фазу инициализации и фазу оптимизации.

Фаза инициализации формирует расписания, руководствуясь только жесткими ограничениями модели. Такие расписания могут быть не оптимальны, но ключевым моментом на данном этапе является получение валидных расписаний.

Фаза оптимизации сосредоточена на улучшении расписания в том числе с учетом удовлетворения мягким ограничениям. Для этого используются операции преобразования расписания, позволяющие изменить время или аудиторию проведения занятия. Важным свойством таких операций является сохранение валидности исходного расписания.

Алгоритмы первой фазы. Алгоритм Any-Fit

Алгоритм Any-Fit, реализующий специфическую задачу упаковки. Первым этапом работы генетического алгоритма является генерация начальной популяции. На этом этапе создается N индивидов (расписаний) с различным генотипом. В классическом варианте генетического алгоритма все особи начальной популяции генерируются полностью случайным образом. Однако, говоря о задаче составления расписания, важно отметить, что полностью случайное формирование генотипа особи (расположение занятий в расписании) не может быть использовано, так как случайная расстановка занятий неизбежно приведет к появлению конфликтов и нарушению ограничений модели, в том числе и жестких ограничений.

Для решения данной проблемы вводится понятие «карта расписания». Последняя отслеживает размещение занятий, объединяющих такие сущности, как преподаватель, студенческая группа и аудитория, по времени в пределах единого для подразделения вуза расписания. Карта представляет собой таблицу логических значений. Всего карты три: для отслеживания размещения занятий для преподавателей, групп и аудиторий.

Размещение занятий происходит случайным образом, с целью формирования разнообразия генотипов в начальной популяции. Карты позволяют обозначать области, свободные для размещения последующих занятий. При этом учитывается занятость групп, преподавателей и аудиторий, периодичность занятий, участие в потоках, требования по оборудованию и прочие ограничения. Таким образом, при помощи карт расписаний строится начальная популяция особей, содержащих только корректные с точки зрения модели расписания.

Алгоритмы первой фазы. Алгоритм с использованием SMT-решателя

SMT-решатель – это программный инструмент, который позволяет решать задачи разрешимости для логических формул с учетом ограничений, определенных в терминах различных теорий. SMT-решатели используются в различных областях, таких как верификация программного обеспечения, автоматическое доказательство теорем, проектирование электронных схем и других.

Преимуществом применения SMT-решателей при такой постановке задачи является то, что ограничения модели уже заданы в виде предикатов. В таком случае, для реализации алгоритма фазы инициализации расписаний с использованием SMT-решателя необходимо всего лишь представить ограничения модели в виде набора переменных и утверждений, которыми сможет оперировать SMT-решатель. В данной работе использовался разработанный компанией Microsoft SMT-решатель Z3 Theorem Prover [10].

Алгоритмы второй фазы. Генетический алгоритм

Фаза оптимизации генетического алгоритма состоит из последовательных операций скрещивания, мутации, отбора и оценки индивидов популяции.

Скрещивание: процесс скрещивания особей в генетическом алгоритме предназначен для увеличения числа особей с генами, дающими лучшее значение оценки. В классическом генетическом алгоритме новая особь наследует половину генетического материала от каждого из родителей. Однако при решении задачи составления расписания подобного подхода сложно добиться из-за требования поддержания корректности расписания. По этой причине кроссинговер реализован следующим образом: одна из родительских особей клонируется и становится новой особью. Для половины случайных генов новой особи по возможности заимствуется у второго родителя.

Мутации: для избежания «скатывания» к локальному минимуму используется механизм мутаций. Мутации позволяют с некоторой вероятностью изменить значение одного или нескольких генов. Внесение подобного шума позволяет особям находить решения с лучшей функцией оценки и исключает сведение решения к локальному минимуму.

Отбор: в процессе отбора восстанавливается исходное число индивидов в популяции путем отбрасывания наименее приспособленных особей. Для этого особи в популяции сортируются по возрастанию значения функции приспособленности, и часть из наименее приспособленных отбрасывается.

Оценка популяции: на данном этапе происходит оценка пригодности особей популяции. Если достигнут определенный уровень приспособленности популяции, либо прошло определенное количество эпох эволюции

популяции, то работа алгоритма заканчивается. Иначе эпоха повторяется – популяция вновь проходит процессы скрещивания, мутации, отбора.

Тестирование алгоритмов составления расписания

Для проверки корректности описания математической модели и работы алгоритмов составления расписания реализованы программные тесты, проверяющие корректность составления расписания в различных сценариях. Для каждого жёсткого ограничения реализовано несколько сценариев тестирования: ситуации, при которых получение расписания возможно множеством способов, единственным способом, а также сценарии, при которых составление корректного расписания невозможно.

Для тестирования производительности алгоритмов, предложены бенчмарк-тесты, оценивающие скорость составления расписаний различных размерностей и конфигураций (см. табл. 1).

Таблица 1

Тесты производительности алгоритмов первой фазы

Количество групп студентов	Длительность семестра, недель	Время генерации расписания для Any-Fit алгоритма, с	Время генерации расписания для алгоритма с использованием SMT-решателя Z3, с
4	4	0,72	0,67
	8	1,32	0,65
	16	1,92	0,66
8	8	4,1	2,52
	12	6,7	2,52
	16	10,1	2,52
12	8	9,31	5,49
	12	13,63	5,54
	16	19,07	5,49
24	16	229,45	23,13

Анализируя результаты тестирования производительности алгоритмов первой фазы, можно сделать вывод, что использование алгоритма с использованием SMT-решателя предпочтительнее, так как он показывает значительный выигрыш во времени, особенно с ростом сложности задачи, не обладая, кроме того, недостатками Any-Fit алгоритма.

Заключение

Разработана математическая модель расписания – совокупность объектов (сущностей) и отношений между ними, отражающими требования к расписанию в форме жёстких и мягких ограничений. Предложенная модель порождает набор алгоритмов, использующих комбинаторные подходы, применение эволюционных теорий, и инструментарий в форме SMT-решателей.

Предложенный подход к построению расписания занятий в вузе показал свою эффективность в части составления начального «неплохого» расписания, которое может быть отправным моментом для составления оптимального расписания, поиск которого осуществляется с помощью генетического алгоритма.

Список литературы

1. Автоматизированное составление расписания. Университет [Электронный ресурс] // solutions.1c.ru [сайт]. – URL: https://solutions.1c.ru/catalog/asp_univer/features (дата обращения: 10.06.2024).
2. Галактика РУЗ [Электронный ресурс] // galaktika.ru [сайт]. – URL: <https://galaktika.ru/ruz> (дата обращения: 10.06.2024).
3. Кабальнов Ю. С., Шехтман Л. И., Низамова Г. Ф., Земченкова Н. А. Композиционный генетический алгоритм составления расписания учебных занятий // Вестник УГАТУ. – 2006. – Т. 7, № 2 – С. 99–107.
4. Маслов М. Г. Эвристический алгоритм решения задачи составления расписания учебных занятий в вузе // Математические методы в технике и технологиях: Сб. трудов XV Международной научной конференции, г. Тамбов, 2–4 июня 2002 г. В 10-ти т. – Тамбов, 2002. – Т. 9. – С. 86–88.
5. Приложение «АВТОРасписание» [Электронный ресурс] // www.utelksp.ru [сайт]. – URL: <http://www.utelksp.ru/programs/prilozhenie-avtoraspisanie> (дата обращения: 10.06.2024).
6. Семенов С. П., Татаринцев Я. Б. Сравнительный анализ подходов к автоматизации составления расписаний учебных занятий в образовательных учреждениях // Известия Алтайского государственного университета. – 2010. – № 1-1. – С. 103–105.
7. Сидорин А. Б., Ликучева Л. В., Дворянкин А. М. Методы автоматизации составления расписания занятий часть 1. Классические методы // Известия ВолгГТУ. – 2009. – Т. 2. – №. 3. – С. 116.
8. Экспресс-расписание вуз [Электронный ресурс] // pbprog.ru [сайт]. – URL: <https://pbprog.ru/docs/raspisv/> (дата обращения: 10.06.2024).
9. Gunawan S., Farhang-Mehr A., Azarm S. Multi-level multi-objective genetic algorithm using entropy to preserve diversity // Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization (EMO 2003), Faro, Portugal. – Lecture Notes in Computer Science – Springer, 2003. – Vol. 2632. – Pp. 148–161.
10. Programming Z3 [Электронный ресурс] // z3prover.github.io [сайт]. – URL: <https://z3prover.github.io/papers/programmingz3.html> (дата обращения: 10.06.2024).