

А. Н. Безгинов, С. Ю. Трезубов

КОМПЛЕКС АЛГОРИТМОВ ПОСТРОЕНИЯ РАСПИСАНИЯ ВУЗА

ЧАСТЬ 2. СТРУКТУРА ПРЕДСТАВЛЕНИЯ ДАННЫХ И АЛГОРИТМ ПОСТРОЕНИЕ ОПОРНОГО РЕШЕНИЯ*

Предложен алгоритм построения множества допустимых расписаний, применяемых в качестве опорных решений в общей задаче оптимизации расписания вуза. Для любых начальных условий алгоритм способен построить хотя бы одно допустимое расписание, если оно существует. Вычислительная сложность снижается за счет специально разработанных эвристик и модели представления расписания.

*Часть 1 см.: Вестник Балтийского федерального университета им. И. Канта. 2011. Вып. 5. С. 127 – 135.



An algorithm for building a base multitude of solutions for the university timetabling problem is proposed. An algorithm allowed to get at least one correct timetable if such exists. Computational complexity is reduced with using a special set of heuristics and timetable data presentation model.

Ключевые слова: построение расписания вуза, начальное расписание, эвристический алгоритм.

Key words: university course timetabling, base multitude of solutions, heuristic algorithm.

Введение

Большинство эвристических алгоритмов поиска оптимального расписания – итерационные (см., например, [1; 2]). Процесс поиска в них – это движение от одного допустимого расписания к другому в направлении улучшения его качества. Отправная точка такого поиска – опорное расписание, результат – расписание, качество которого улучшить нельзя. Поскольку множество результирующих неулучшаемых расписаний может быть обширно (парето-множество), то с учетом зависимости результата поиска от начального расписания целесообразно в качестве опорного использовать не одно, а множество расписаний. В этом случае на основе каждого элемента рассматриваемого множества опорных расписаний организуется независимый процесс поиска подмножества парето-оптимальных расписаний.

Общая идея подхода к оценке качества и построения алгоритма поиска оптимального расписания была рассмотрена нами в статье [3]. В данной работе предлагается алгоритм построения одного или множества опорных расписаний. В разработке алгоритма выделим две части:

- разработку структуры представления данных, учитывающую большинство ограничений, накладываемых на расписание;
- разработку алгоритма построения опорного решения, результатом работы которого всегда является хотя бы одно допустимое расписание, если оно существует в рамках заданных начальных условий.

1. Структура представления данных

При разработке структуры представления данных помимо группирования задействованных в расписании ресурсов была поставлена цель – учесть в структуре взаимосвязи между рассматриваемыми ресурсами и отношения предпочтения каждого участника расписания на множестве его индивидуальных расписаний. Для упрощения этой задачи и алгоритма в целом построение допустимого расписания занятий для вуза декомпозировано на две подзадачи меньшей размерности:

- 1) распределение занятий, предписанных преподавателям и группам учебным планом, во временной сетке вуза с учетом емкости имеющегося аудиторного фонда для занятий различного типа;



2) назначение для занятий конкретных аудиторий (такое назначение всегда возможно, так как распределение занятий во временной сетке вуза учитывает емкость его аудиторного фонда).

Введем понятия и обозначения, принятые в данной работе.

Пусть P и G — множества преподавателей и групп рассматриваемого учебного заведения, а D_i — множество дисциплин, сопоставленных учебным планом G_i -й группе. Введем понятие «занятие». Под занятием L_{ijk} будем понимать аудиторную пару, проводимую преподавателем $P_k \in P$ в группе $G_i \in G$ по дисциплине $D_{ij} \in D$.

Вместо группы в занятие может входить поток (объединение групп) или подгруппа. Такое расширение несколько усложняет рассматриваемую структуру, не изменяя ее сущности и алгоритмов работы с ней, поэтому далее в данной работе не рассматривается.

Будем рассматривать расписание на неделю. Если у некоторой группы по некоторой дисциплине предусмотрено несколько пар в неделю, то рассматриваем их как разные занятия.

Рабочую неделю вуза представим множеством аудиторных пар T_A . Каждый преподаватель P_k и группа G_i имеет свое рабочее расписание, которое является подмножеством T_A .

Рассмотрим структуру представления данных для первой подзадачи — распределение занятий во временной сетке вуза (структура представления данных для второй подзадачи строится аналогично).

Отобразим множество занятий L_{ijk} на расписания групп G_i , вовлеченных в эти занятия. Для этого свяжем каждое занятие с множеством пар в расписании соответствующей группы, допустимых рабочим расписанием преподавателя и группы. В результате связи, относящиеся к одному занятию, образуют группу всех возможных вариантов размещения данного занятия в расписании. В процессе решения задачи среди всех вариантов должен быть выбран только один.

Каждой связи припишем вес W_{ijkt} , где $t \in T_A$ — соответствующая пара в расписании. Вес W_{ijkt} — это численное выражение степени удовлетворенности требований задействованных в занятие L_{ijk} участников расписания в случае проведения рассматриваемого занятия на данной паре, учитывающее сложную структуру взаимосвязей между преподавателями и группами. Он определяется по следующим формулам:

$$W_{ijkt} = ds_{ikt} \cdot (1 - dc_{ikt}) \cdot dr_i \cdot rc_{ijk}; \quad (1)$$

$$dc_{ikt} = (MC_{ikt} - MinC_{ikt}) / (MaxC_{ikt} - MinC_{ikt}); \quad (2)$$

$$MC_{ikt} = \min_{l \in \{LS_{ikt}\}} \{C_l\}, \quad MinC_{ikt} = \min_{t \in \{T_A\}} \{MC_{ikt}\}, \quad MaxC_{ikt} = \max_{t \in \{T_A\}} \{MC_{ikt}\},$$

где C_l — число вариантов размещения занятия $l \in L_{ijk}$ в сетке расписания; LS_{ikt} — множество занятий ($LS_{ikt} \subseteq \{L_{ijk}\}$), которые потенциально могут быть назначены на t -ю пару и в которых задействован либо преподаватель P_k , либо группа G_i ; ds_{ikt} — численное выражение степени удовле-



творности требований задействованных в данном занятии преподавателя P_k и группы G_i в случае проведения занятия на t -й паре; dr_t — доля доступного (не задействованного в других занятиях) аудиторного фонда, который потенциально может быть задействован на t -й паре для проведения занятия L_{ijk} в общем объеме аудиторного фонда группы, доступного для проведения данного занятия (значения dr_t нормируются в соответствии с условием $\sum_{t \in T_{ik}} dr_t = 1$, $T_{ik} = T_i \cap T_k$ — множество пар, доступных для проведения рассматриваемого занятия L_{ijk}); rc_{ijk} — величина, обратная мощности множества вариантов проведения занятия L_{ijk} .

Поясним дополнительно введенную выше величину ds_{ikt} . Для этого рассмотрим некоторое занятие L_{ijk} в котором задействованы преподаватель P_k и группа G_i , имеющие расписания $T_k \subseteq T_A$ и $T_i \subseteq T_A$ соответственно, а также множество пар $T_{ik} = T_i \cap T_k$ доступных для проведения рассматриваемого занятия. Преподаватель и группа (в качестве защитника ее интересов выступает диспетчер) могут иметь свои предпочтения относительно времени проведения данной дисциплины, такие как: желание преподавателя сконцентрировать свои занятия в конкретные дни (пары), требование проведения наиболее сложных для усвоения дисциплин в середине недели и т. п. Заметим, что поскольку на этом этапе расписание еще не сформировано, то рассматриваются только требования, не учитывающие взаимное расположение занятий в расписании. На основе данных требований с использованием системы оценки расписания, общие принципы построения которой описаны в работе [3], каждой паре $t \in T_{ik}$ ставится в соответствие величина ds_{ikt} характеризующая степень выполнения указанных требований в случае проведения занятия на данной паре. Для каждого занятия полученные значения ds_{ikt} нормируются в соответствии с условием: $\sum_{t \in T_{ik}} ds_{ikt} = 1$.

Использование рассчитанных таким способом весов W_{ijkt} позволяет существенно сократить время построения множества опорных расписаний.

Таким образом, рассмотренная структура описывает набор вариантов размещения занятий в расписании вуза. Для того чтобы в данной структуре выделить конкретное расписание, необходимо для каждого занятия выделить один вариант его размещения в расписании. Связи, реализующие такой вариант, будем называть *активными*.

Чтобы полученное расписание удовлетворяло условиям допустимости, необходимо выполнение следующих жестких требований (в скобках дается эквивалентная формулировка «на языке связей»):

— в каждой группе должны быть проведены все запланированные занятия (каждому занятию должна быть сопоставлена ровно одна активная связь);

— в каждой группе в каждый момент времени может проводиться не более одного занятия (в расписании каждой группы каждой паре может быть сопоставлена только одна активная связь);



— каждый преподаватель в каждый момент времени может вести не более одного занятия (каждому преподавателю на каждой паре сопоставляется только одна активная связь);

— множество занятий, проводящихся в каждый момент времени, не должно выходить за рамки имеющегося объема аудиторного фонда (множество занятий, установленных в расписании на данной паре с помощью активных связей, не должно выходить за рамки имеющегося объема аудиторного фонда).

Построенная таким образом структура представления данных учитывает все накладываемые на задачу ограничения, за исключением тех, в которых учитывается взаимное расположение различных дисциплин в готовом расписании. Некоторые аспекты рассмотренной структуры были детально проанализированы в работе [4].

2. Алгоритм формирования опорного расписания

В данном разделе рассмотрен алгоритм построения множества опорных расписаний на основе структуры, описанной в предыдущем разделе. Алгоритм полный, то есть гарантирует получение хотя бы одного допустимого расписания, если оно существует.

Алгоритм построения одного опорного расписания является итерационным. Работа алгоритма начинается с пустого расписания. На каждой итерации в текущее расписание добавляется очередное занятие, при этом сохраняется корректность (допустимость) расписания. Данный процесс продолжается до тех пор, пока все занятия не будут размещены в расписании либо не будет показано, что такого размещения не существует. Рассмотрим основные элементы алгоритма.

Шаг 1. Строим описанную в предыдущем разделе структуру представления данных для обеих подзадач из числа выделенных в начале раздела 1 без указания активных связей. Рассчитаем в этих структурах вес, соответствующий каждой связи, а также доступный на каждой паре объем аудиторного фонда для занятий каждого типа.

Далее на шагах 2–4 рассматривается алгоритм распределения занятий во временной сетке вуза, то есть решения первой подзадачи. Алгоритм распределения занятий по аудиториям (вторая подзадача) строится аналогично.

Шаг 2. Если нераспределенных занятий нет, то алгоритм заканчивает работу. Если же ситуация иная, то среди всех доступных связей выберем активную (указание для данного занятия активной связи эквивалентно размещению его в расписании). Правила выбора активной связи таковы:

1) если занятие имеет только один вариант размещения в расписании, то связь, реализующая данный вариант, становится претендентом на активную связь;

2) если найдена только одна связь — претендент на активную, назначим ее активной и переходим к шагу 3. Если таких связей несколько, то проверим их на совместность. Связи будем называть совместными, если назначение их активными не приводит к нарушению жестких требова-



ний задачи. В случае совместности полученных связей назначаем их активными и переходим к шагу 3, в противном случае — к шагу 4;

3) если в предыдущих пунктах ни одной активной связи найти не удалось, то в качестве активной назначим связь с наибольшим весом и переходим к шагу 3.

Шаг 3. Исключим из дальнейшего рассмотрения все связи, противоречащие установленным активным связям. Вследствие этого изменятся веса некоторых связей; пересчитаем их и перейдем к шагу 2.

Шаг 4. Занятия могут оказаться несовместными, если для них необходимы ресурсы, использованные занятиями, уже размещенными в расписании. Для того чтобы такие занятия сделать совместными, необходимо высвободить нужные ресурсы по следующему алгоритму:

1) составим список размещенных в расписании занятий, использующих недостающие ресурсы, и рассмотрим занятие из списка, размещенное в расписании последним;

2) высвободим необходимый для несовместных занятий ресурс, с этой целью укажем рассматриваемому занятию альтернативную активную связь. Если такой связи не существует (то есть других связей нет или они заблокированы), то рассмотрим следующее занятие из списка и перейдем к началу пункта 2 данного шага. Если все занятия из списка рассмотрены, то попытаемся на основе аналогичного алгоритма высвободить ресурсы для перемещения активных связей занятий из рассматриваемого списка. Будем повторять данную процедуру до тех пор, пока не высвободим необходимые ресурсы для несовместных занятий или не покажем, что это сделать невозможно (для этого потребуются последовательно перебрать все возможные варианты его освобождения), и тогда алгоритм завершает работу;

3) если необходимый ресурс найден, то укажем для рассматриваемых занятий соответствующие активные связи и перейдем к шагу 3.

Следует отметить, что вес связи не только является выражением степени удовлетворения требований участников расписания, но и учитывает сложную структуру взаимосвязей между преподавателями и группами. Так как в алгоритме построения опорного расписания на этапе выбора активной связи при равных условиях выбирается связь с наибольшим весом, то предпочтение отдается связям, задействующим наименее загруженные пары в расписании преподавателя и группы, и занятиям, имеющим наименьшее число вариантов размещения в расписании. Это следует из анализа формул (1) и (2) для расчета веса связи. Благодаря этому в алгоритме существенно сокращается число возвратов для освобождения уже занятого ресурса.

Описанный алгоритм строит одно опорное расписание. Для получения множества опорных расписаний необходимо реализовать распараллеливание процесса построения расписания. Для этого на этапе выбора активной связи в случае равенства или близости весов нескольких связей на основе данной структуры строятся две новые. Для этих двух структур процессы построения продолжают независимо друг от друга. После завершения работы алгоритма из множества полученных расписаний отбирается нужное количество наиболее непохожих — они и составляют опорное множество допустимых расписаний.



3. Особенности реализации предложенного алгоритма в программном комплексе составления расписания

Предложенный в разделе 2 алгоритм построения опорного расписания основан на последовательном размещении очередного занятия в сетке расписания и процедуре высвобождения недостающего ресурса для очередного занятия путем перераспределения ресурсов между уже размещенными в расписании занятиями. Важной характеристикой подобного рода алгоритмов является скорость нахождения итогового решения. В данном случае скорость напрямую зависит от числа итераций, необходимых алгоритму для размещения всех занятий в сетке расписания. Очевидно, что минимальное возможное число итераций равно мощности исходного множества занятий. Число итераций может увеличиваться за счет вызова процедуры перераспределения ресурсов. Таким образом, более эффективным естественно считать алгоритм, вызывающий процедуру перераспределения ресурсов меньшее число раз.

В предложенном алгоритме минимизация числа вызовов процедуры перераспределения ресурсов достигается за счет введения приоритетов на множестве вариантов размещения занятий в расписании. Приоритеты, в свою очередь, задаются весами (см. (1)).

Для демонстрации эффективности алгоритма была разработана комплексная программа, автоматизирующая процесс составления расписания. В программе, помимо предложенного (основного) алгоритма построения опорного расписания, был реализован упрощенный алгоритм, в котором на каждой итерации очередное занятие и его позиция в расписании выбираются без учета приоритетов отдельных вариантов (то есть все варианты равноправны). Для программы было составлено множество тестовых примеров, охватывающих широкий класс задач составления расписания различной размерности. На базе каждого тестового примера были проведены эксперименты с обоими алгоритмами — основным и упрощенным.

Полученные результаты позволили сделать следующие выводы. Основной алгоритм по сравнению с упрощенным уменьшил число вызовов процедуры перераспределения ресурсов в среднем более чем в 4 раза, благодаря чему время, необходимое для построения одного опорного расписания, сократилось более чем в 2 раза. Заметим, что указанный порядок прироста эффективности алгоритма является средним показателем для использованного множества тестовых задач. В ряде тестовых задач, как правило, отличающихся особой плотностью расписания, время построения опорного расписания было сокращено более чем в 10 раз.

Дополнительно исследованы возможности повышения качества опорного расписания. Для этого использовалась средневзвешенная оценка штрафов за полное или частичное невыполнение каждого из требований, предъявляемых к расписанию. Как пути повышения качества расписания рассматривались варианты модификации описанного



алгоритма через добавление в него новых эвристик, стремящихся минимизировать штрафы за невыполнение отдельных требований.

Заметим, что многие требования к расписанию (например, минимизировать количество окон в расписании группы) задают ограничение на расположение занятий относительно друг друга в итоговом расписании. Этот факт затрудняет разработку действенных эвристик, минимизирующих штрафы для подобных требований на этапе построения опорного расписания.

С учетом вышесказанного, а также противоречивости, присущей требованиям различных субъектов — участников учебного процесса, было принято решение на этапе построения опорного расписания рассматривать не все, а только наиболее принципиальные требования к расписанию. В их число вошли: требования «минимизировать число окон в расписании группы (преподавателя)» и «распределять учебную нагрузку группы (преподавателя) равномерно в течение недели».

Комбинация данных требований позволяет компенсировать неоднозначность, возникающую на этапе построения расписания при оценке степени выполнения рассматриваемых требований. Для демонстрации неоднозначности рассмотрим пример. Пусть в расписании группы G_i нужно разместить очередное занятие L . Если L не является последним неразмещенным в расписании группы занятием, то на данном этапе сложно с полной уверенностью судить о наличии или отсутствии окон в итоговом расписании для D_i , так как позиции других неразмещенных занятий остаются неопределенными.

Рассмотрим требование «распределять учебную нагрузку группы G_i равномерно в течение недели». Так как общее число занятий GLC_i в расписании группы и число ее учебных дней GDC_i заранее известны из учебного плана, то для G_i можно определить максимальное допустимое число занятий в день $MGLC_i$ по формуле:

$$MGLC_i = GLC_i \operatorname{div} GDC_i + (GLC_i \bmod GDC_i > 0? 1: 0).$$

Аналогичным образом определяется $MPLC_k$ — максимальное допустимое число занятий в день для преподавателя P_k .

На основе предложенных рассуждений сформулируем следующую дополнительную эвристику: при размещении в расписании на некоторой паре p в день d очередного занятия L , в котором задействованы группа G_i и преподаватель P_k , расстояние между p и первой (последней) парой в данный день d в расписаниях G_i и P_k не должно превышать $MGLC_i$ и $MPLC_k$ соответственно.

Под расстоянием между двумя парами здесь понимается число занятий, которое потенциально может быть размещено на этих парах и во временном интервале между ними.

На практике добавление предложенной эвристики в алгоритм построения опорного расписания увеличило возможности по удовлетворению выбранных высокоприоритетных требований в итоговом опорном расписании. Однако при этом в несколько раз возросло число вызовов процедуры перераспределения ресурсов. Это обусловлено тем,



что учет предложенной эвристики существенно сокращает множество допустимых вариантов размещения очередного занятия в сетке расписания. Поэтому для рассматриваемой эвристики был разработан следующий ослабленный вариант: если на занятие L наложены требования «минимизировать число окон в расписании группы (преподавателя)» и «распределять учебную нагрузку группы (преподавателя) равномерно в течение недели», то величины весов W_{ijkl} связанных с данным занятием и соответствующих позициям, удовлетворяющим исходной эвристике, увеличиваются в $\sim 1,1$ раза (это значение получено эмпирически).

На практике применение ослабленного варианта эвристики позволило в среднем повысить качество опорного расписания на 10 %, при этом число вызовов процедуры перераспределения ресурсов увеличилось только на 4 %.

Анализ эффективности описанного в данной статье алгоритма построения опорного расписания проводился на реальных задачах составления расписания для филиала ГОУ МГИУ в г. Сергиев Посад. В филиале 29 групп студентов, обучающихся по четырем специальностям, 73 преподавателя и 17 аудиторий. На базе этого вуза было разработано и решено множество тестовых задач составления расписания, отличающихся друг от друга набором предъявляемых к расписанию требований. Проведенные вычислительные эксперименты показали, что алгоритму для построения допустимого расписания филиала ГОУ МГИУ в среднем требуется от двух до пяти минут вычислительного времени на компьютере с процессором AMD Athlon64 3800+.

Заключение

В работе предложена структура представления данных и алгоритм поиска опорного множества расписаний. Алгоритм всегда находит хотя бы одно допустимое расписание, если оно существует. При этом свойственная подобным алгоритмам вычислительная сложность снижается за счет применения весов, позволяющих при последовательном наполнении опорного расписания конкретными занятиями задействовать, прежде всего, дефицитные ресурсы, то есть специализированные аудитории и наиболее загруженных преподавателей.

Были исследованы возможности повышения качества опорного расписания путем добавления в предложенный алгоритм дополнительной эвристики, направленной на минимизацию штрафов за невыполнение отдельных требований. Для двух высокоприоритетных требований была разработана специальная эвристика. Ее применение исследовано на множестве тестовых примеров, показавших эффективность этой эвристики.

Рассмотренные в данной работе идеи построения алгоритма поиска опорного решения в рамках комплексного алгоритма оптимизации расписания могут быть использованы в системах формирования расписаний для других предметных областей, в которых предполагается



распределение разнородных ресурсов в пространственно-временной сетке функционирования сложной системы с учетом нечетких и противоречивых требований субъектов — участников процесса, регулируемого расписанием.

Список литературы

1. *Burke E., Kingston J., Jackson K., Weare R.* Automated university timetabling: the state of the art // *The Computer Journal.* 1997. Vol. 40, №9. P. 565–571.

2. *Muller T.* Some novel approaches to lecture timetabling, Charles University // *Constraint Programming for Decision and Control.* 4th CPDC'2002, Poland. 2002. P. 31–37.

3. *Безгинов А. Н., Трегубов С. Ю.* Комплекс алгоритмов построения расписания вуза. Ч. 1: Система оценки качества расписания на основе нечетких множеств, алгоритм поиска оптимального расписания // *Вестник Балтийского федерального университета им. И. Канта.* 2011. Вып. 5. С. 127–135.

4. *Безгинов А. Н., Трегубов С. Ю.* Об одном методе формирования расписания // *Информационные технологии и программирование: межвузовский сборник статей.* Вып. 2 (16). М., 2006. С. 5–13.

Об авторах

Анатолий Николаевич Безгинов — д-р техн. наук, проф., филиал Московского государственного индустриального университета, г. Сергиев Посад Московской обл., e-mail: mail@spf-mgiu.ru.

Сергей Юрьевич Трегубов — ассист., филиал Московского государственного индустриального университета, г. Сергиев Посад Московской обл., e-mail: s.tregubov@gmail.com.

Authors

Prof. Anatoliy Bezginov — Branch of Moscow state industrial university, Sergiev Posad city Moscow region, e-mail: mail@spf-mgiu.ru.

Sergey Tregubov — assistant, Branch of Moscow state industrial university, Sergiev Posad city Moscow region, e-mail: s.tregubov@gmail.com.