УДК 519.8

МОДЕЛИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ ЗАДАЧИ СОСТАВЛЕНИЯ РАСПИСАНИЯ УЧЕБНЫХ ЗАНЯТИЙ

И. Л. Каширина, А. Л. Ухин

Воронежский государственный университет,
Воронежский государственный архитектурно-строительный университет

Поступила в редакцию 16.02.2015 г.

Аннотация. В статье рассматривается объектный подход к моделированию процесса составления учебного расписания, позволяющий понизить размерность решаемой задачи и разработать для нее эффективный генетический алгоритм. Статья содержит подробное описание всех этапов генетического алгоритма с учетом возможного использования дополнительных опций, улучшающих качество расписания. В заключении приводятся результаты вычислительного эксперимента.
Ключевые слова: генетический алгоритм, квадратичная задача о назначениях, расписание учебных занятий, критерии качества расписания.
Annotation. The article deals with the object approach to modeling the process of drawing up a training schedule that allows you to reduce the dimension of the problem being solved and provide it with an efficient genetic algorithm. The article contains a detailed description of all phases of the genetic algorithm, taking into account the possible use of additional options that improve the quality of the schedule. In conclusion, the results of computational experiments.
Keywords: genetic algorithm, quadratic task assignments, schedule training sessions, quality criteria schedule.

ВВЕДЕНИЕ

Проблеме автоматизации составления учебных расписаний посвящено большое количество работ, различающихся в зависимости от требований образовательных структур [5–8]. Формально задача составления учебного расписания относится к классу задач нелинейного целочисленного программирования, как правило, большой размерности. Целевая функция такой задачи обычно предполагает минимизацию суммарных штрафов за нарушение желательных, но не обязательных свойств расписания, а в качестве ограничений формализуются требования, которые обязательно должны быть выполнены. Точное решение такой задачи практически невозможно получить за приемлемое время в силу ее большой размерности (десятки тысяч переменных), поэтому на практике прибегают к приближенным поисковым методам.

В данной статье предлагается один из возможных вариантов постановки задачи составления учебного расписания и метод ее решения, основанный на использовании генетического алгоритма. Учебное расписание является довольно сложной системой, все элементы которой взаимосвязаны между собой напрямую или косвенно. Помимо большого количества условий, которые приходится строго выполнять, необходимо оптимизировать множество параметров, влияющих на качество расписания. При этом очень трудно получить результат, который учитывал бы все особенности конкретного учебного заведения и удовлетворял обязательным санитарно-эпидемиологическим требованиям к условиям и организации обучения, представленных, в частности, рядом требований к расписанию занятий [4]. Предлагаемый в статье генетический алгоритм, хотя и является в общем случае приближенным, позволяет выполнить все обязательные условия, и показывает высокую эффективность в ходе вычислительного эксперимента.

ПОСТАНОВКА ЗАДАЧИ

Исходными данными в задаче составления расписания учебных занятий являются:

1) множество преподавателей P;

2) множество учебных групп K;

3) множество учебных дисциплин (предметов) для каждой группы U;

4) множество аудиторий A;

5) учебный план, указывающий каждой учебной группе количество часов по каждому из изучаемых предметов.

Необходимо построить расписание, обеспечивающее проведение всех занятий без накладок относительно аудиторий и без «окон» для обучающихся, удовлетворяющее требованиям СанПиН [4], а также, по возможности, учитывающее пожелания преподавателей.

За основу для моделирования выбирается объектный подход [2]. В качестве основного элемента моделирования используется класс "Предмет", объекты которого обладают следующими свойствами: название предмета, учебная группа, в которой он преподается, ФИО преподавателя, номера аудиторий, в которых должен преподаваться этот предмет (либо указание, что специализированные аудитории для него отсутствуют). Все объекты-предметы нумеруются в произвольном порядке числами от 1 до n, причем количество экземпляров каждого объекта, участвующих в нумерации, равно количеству часов в неделю данного предмета в соответствующей учебной группе. Расписание представляет собой таблицу, ячейки которой также пронумерованы специальным образом. Предварительно фиксируется структура будущего расписания, то есть указываются ячейки, которые не будут участвовать в построении расписания. Если структура расписания будет жестко зафиксирована (то есть количество ячеек, участвующих в расписании, окажется в точности равно количеству учебных часов в неделю в данной группе) – это будет гарантировать отсутствие окон у обучающихся.

Далее нумерация ячеек таблицы происходит по следующему принципу:

$$\left[\begin{pmatrix}1\\ \vdots\\ \vdots\\ g_1\end{pmatrix}, \begin{pmatrix}g_1+1\\ \vdots\\ \vdots\\ g_2\end{pmatrix}, \ . \ . \ . \ , \begin{pmatrix}g_{q-1}+1\\ \vdots\\ \vdots\\ m\end{pmatrix}\right], \quad (1)$$

где q – общее количество учебных групп, участвующих в расписании, $(g_j - g_{j-1})$ – количество учебных часов в неделю, проводимых согласно учебному плану в j-ой группе, m – общее количество «свободных» ячеек. Пример таблицы расписания (на один день для 4-х групп) с пронумерованными ячейками и зафиксированной структурой изображен на рис. 1.

Номер урока	*Номер группы*			
	1	***2***	***3***	***4***
1	1	4	8	
2	2	5	9	13
3	3	6	10	14
4		7	11	15
5			12	16

Рис. 1. Пример зафиксированной структуры расписания

Заштрихованные серым цветом ячейки не участвуют в построении расписания. То есть в данном случае в группе 1 предусмотрено не более 3-х уроков в день, группа 4 начинает занятия со 2-го урока и т.д.

Далее для каждого объекта-предмета под номером k вводится в рассмотрение еще одно свойство: множество $I(k)$ номеров строк расписания, на которых может находиться данный предмет (в соответствии с учебным планом и пожеланиями преподавателя, который его ведет). Соответственно, могут быть сформированы множества $K_j(i)$ – номера предметов j-ой уч. группы, которые могут быть стоять в i-й строке расписания.

Тогда расписание может быть представлено в следующем виде:

$$Y = \left(y_1, \ \ldots, y_q\right) = \left[\begin{pmatrix}y_{11}\\ \vdots\\ \vdots\\ y_{p1}\end{pmatrix}, \ . \ . \ . \ , \begin{pmatrix}y_{1q}\\ \vdots\\ \vdots\\ y_{pq}\end{pmatrix}\right]. \quad (2)$$

Здесь Y – это матрица размера $p \times q$, y_j – вектор-расписание, соответствующее

j-ой учебной группе, p – общее количество строк в таблице-расписании, при этом $y_{ij}=k$, если в j-ой учебной группе на i-той строке расписания находится предмет k, и $y_{ij}=0$, если в j-ой учебной группе i-тая строка пустая.

1	4	12	0
3	5	11	16
2	7	8	15
0	6	9	14
0	0	10	13

Рис. 2. Фрагмент одного из возможных расписаний

Рассмотрим матрицу совместимости предметов S, которая вводится следующим образом:

$$S(k,r)=\begin{cases}0, & \text{если предмет } k \\ & \text{совместим с предметом } r \\ 1, & \text{иначе,} \quad r=\overline{1,n},\ k=\overline{1,n}.\end{cases}$$

Предметы являются несовместимыми, если они используют общий ресурс (преподавателя или аудиторию). Такие предметы не могут находиться в расписании на одной строке (то есть преподаваться одновременно).

Несовместимость между предметами, проводимыми в одной учебной группе можно не учитывать, так как в матрице y_{ij} они находятся в одном столбце, соответственно, автоматически не могут находиться на одной строке.

Целевая функция минимизирует количество несовместимых предметов в каждой строке:

$$\sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{l=j+1}^{q} S(y_{ij},y_{il}) \to \min. \qquad (3)$$

Очевидно, что оптимальное значение данной функции равно 0. Достижение этого предела является обязательным условием для получения приемлемого расписания. Эта функция будет в дальнейшем использована при вычислении критерия приспособленности генетического алгоритма, который работает именно с переменными y_{ij}.

Для получения строгой математической постановки задачи введем дополнительно бинарные переменные:

$$z_{ki}=\begin{cases}1, & k\text{-ый предмет преподается} \\ & \text{на } i\text{-ой строке} \\ 0, & \text{иначе;} \qquad k=\overline{1,n} \quad i=\overline{1,p}\end{cases} \qquad (4)$$

Ограничения, определяющие допустимое множество:

$$\sum_{i\in I(k)} z_{ki}=1,\ z_{ki}=0,\ \ i\notin I(k),\ k=\overline{1,n} \qquad (5)$$

(каждый предмет k проводится только один раз, причем на допустимой строке расписания);

$$\sum_{k\in K_j(i)} z_{ki}=1, i=\overline{1,p}, j=\overline{1,q} \qquad (6)$$

(в каждой группе j на каждой «разрешенной» строке i проводится только один предмет);

$$\sum_{i=1}^{p}\sum_{j=1}^{q}\sum_{g_{j-1}<k\le g_j}\sum_{l>g_j} S_{kl}\, z_{ki} z_{li} \to \min \qquad (7)$$

(целевая функция минимизирует количество несовместимых предметов, преподающихся на каждой строке).

Задача (4)–(7) является квадратичной задачей о назначениях, некоторые алгоритмы решения которой представлены в [1–2], причем размерность этой задачи составляет около 1000 переменных, что существенно меньше размерности большинства известных моделей задач о составлении учебного расписания.

С учетом введенных переменных расписание теперь можно представить в виде, изображенном на рис. 3.

В первом столбце перечислены предметы каждой учебной группы. Серым цветом отмечены запрещенные места для проведения k-го предмета на i-ом уроке. Единица в ячейке таблицы означает, что данный предмет проводится на данном уроке. Такое представление расписания наглядно показывает допустимое множество задачи (4)–(7) и позволяет легко построить начальные допустимые точки для последующей инициализации генетического алгоритма. Переход к переменным y_{ij} осуществляется следующим образом: $y_{ij}=k$, если $z_{ki}=1$, $g_{j-1}<k\le g_j$.

ГЕНЕТИЧЕСКИЙ АЛГОРИТМ

Несмотря на то, что с учетом предлагаемого подхода к моделированию размерность задачи составления учебного расписания мо-

	Предмет	Урок 1	Урок 2	Урок 3	Урок 4	Урок 5
гр 1	1	1				
	2			1		
	3		1			
гр 2	4	1				
	5		1			
	6				1	
	7			1		
гр 3	8			1		
	9				1	
	10					1
	11		1			
	12	1				

Рис. 3. Фрагмент одного из расписаний в терминах переменных z_{ki}

жет быть существенно уменьшена, для точных методов эта размерность по-прежнему остается достаточно большой. В связи с этим для решения полученной задачи предлагается использовать генетический алгоритм.

Как известно, общая схема генетического алгоритма может быть представлена следующим образом [3].

1. Формирование начальной популяции.
2. Оценка особей популяции.
3. Отбор (селекция).
4. Скрещивание.
5. Мутация.
6. Формирование новой популяции.
7. Если популяция не сошлась, то 2. Иначе – останов.

На начальном этапе решения задачи в качестве генотипа предлагается рассматривать матрицу $Y = (y_{ij})$. Алгоритм перестраивает ее таким образом, что элементы матрицы будут менять положение только в пределах своего столбца. При кроссовере часть столбцов случайным образом берется от одного родителя, часть от другого. Таким образом, потомки всегда будут представлять допустимые решения. Мутация с некоторой вероятностью меняет местами элементы одного столбца; это случайно выбранные элементы (участвующие в расписании), обмен которых дает допустимое решение. Оценка особей популяции осуществляется с помощью целевой функции (3). Так как эта функция минимизируется, то в качестве критерия отбора целесообразно использовать турнирный отбор.

Однако вычислительный эксперимент показал, что данный алгоритм улучшает расписание до некоторого предела, а далее процесс улучшения замедляется. Точное решение (у которого, как известно, должно быть нулевое значение целевой функции) при этом получить, как правило, не удается. Но за сравнительно небольшое число итераций (500–1000) можно отыскать приемлемое приближенное решение.

С целью улучшения характеристик расписания предлагается использовать следующий подход: на следующем этапе алгоритма оптимизировать расписание каждой отдельно взятой учебной группы. Таким образом, в решении, полученном на 1 этапе, генетический алгоритм применяется далее к каждому отдельно взятому столбцу.

Функция приспособленности для g-го столбца определяется следующим образом:

$$\sum_{i=1}^{p} \sum_{j<>g} S(y_{ij}, y_{ig}) \to \min, \qquad (8)$$

(на каждой строке i подсчитывается кол-во предметов, несовместимых с предметом y_{ig} и результат суммируется). Размерность «эволюционирующего» вектора теперь не так велика, и с помощью генетического алгоритма можно довольно быстро найти нулевое (или близкое к нулю) решение данной локальной задачи. Улучшив целевую функцию локальной задачи, мы соответственно улучшаем общую целевую функцию. Обработав все векторы (каждый из которых соответствует определенной учебной группе), процесс целесообразно повторить сначала, т.к. каждый вектор настроился под остальные, а те в свою очередь изменились. Использование принципа элитизма (сохранения в новой популяции наилучшего решения из предыдущей) позволяет получить результат не хуже, чем был получен ранее.

Для решения локальной задачи предлагается использовать оператор скрещивания следующего вида.

5	10	8	3	7	2	4	6	9	1	Родитель 1
5	8	1	2	9	3	7	10	4	6	Родитель 2
5	8	1	2	7	3	4	10	9	6	Потомок 1
5	10	8	3	9	2	7	6	4	1	Потомок 2

Рис. 4. Иллюстрация оператора скрещивания

Генотип родителей разбивается на подмножества, одинаковые для обоих родителей. Например, на рис. 4 это {5}, {8,1,2,3,10,6}, {7,4,9}. Если рассматривать одного родителя как перестановку другого, то мы выделяем независимые циклы в этой перестановке.

Часть таких подмножеств берется от одного родителя, часть от другого. Каждый элемент в генотипе потомка будет при этом находиться на том месте, на котором он находился в одном из родителей, при этом такой кроссовер всегда будет генерировать допустимые решения-потомки.

ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ АЛГОРИТМА

Важной положительной характеристикой предлагаемого алгоритма является простое включение в рассмотрение дополнительных требований, улучшающих качество расписания. Сюда можно отнести:

1. *Возможность не фиксировать четко границы расписания* (например, указывается, что в четверг и в пятницу должно быть как минимум 4 урока, а пятый урок только в какой-то один из этих дней). Для этого добавляется фиктивный предмет, с определенными свойствами, и по окончании работы алгоритма, ячейка с фиктивным предметом будет равнозначна тем, которые не участвуют в расписании.

2. *Наличие спаренных уроков*. Для учета спаренных уроков вводится матрица:

$$M_{ij} = \begin{cases} 1, \text{ уроки } i, j \text{ должны проводиться} \\ \quad \text{непосредствено один за другим,} \\ 0, \text{ иначе,} \end{cases}$$

и подсчитывается количество ситуаций в каждой учебной группе не удовлетворяющих данному условию. Полученное значение добавляется в качестве штрафа в целевую функцию (8) с некоторым весовым коэффициентом. Другой вариант, – изначально создавать популяцию, в которой необходимые уроки находятся рядом. При этом в кроссовере и мутации учитывается, чтобы они не разбивались по ходу алгоритма.

3. *Распределение нагрузки в течение учебной недели*. В соответствии с требованиями СанПиН [4], каждый учебный предмет имеет свою оценку трудности (например, математика – 8 баллов, физическая культура – 1). При этом наибольшая учебная нагрузка (в баллах) должна приходиться на середину недели


Расписание для классов

Класс 5б

Урок	Пн	Вт	Ср	Чт	Пт	Сб
1	Экология	Литература	Ин яз	ИЗО	Ин яз	Литература
2	Русский	Математика	Математика	История	Литература	Физкультура
3	Физкультура	Музыка	Математика	Ин яз	Экология	Русский
4	Русский	История	Русский	Математика	Литература	Технология
5		Русский	Ин яз	Математика	Математика	Технология
6						
Нагрузка	32	38	53	42	40	31

Рис. 5. Фрагмент расписания с учетом распределения нагрузки

(среду и четверг), а последний учебный день (субботу) нагрузка должна быть наименьшей. Штрафы за нарушение указанных условий также могут быть добавлены в целевую функцию (8). На рис. 5 приведено построенное предлагаемым алгоритмом расписание для одного из классов с учетом указанных требований к распределению нагрузки.

ЗАКЛЮЧЕНИЕ

Проведенные вычислительные эксперименты показывают эффективность предлагаемого генетического алгоритма построения расписания. Для всех 100 случайным образом сгенерированных тестовых задач (с различными учебными планами и размерностью 900–1000 переменных) после настройки параметров алгоритма было найдено оптимальное решение с нулевым значение целевой функции (без учета дополнительных штрафных коэффициентов).

СПИСОК ЛИТЕРАТУРЫ

1. *Каширина И. Л.* Генетический алгоритм решения многокритериальной задачи о назначениях при нечетких коэффициентах целевой функции / И. Л. Каширина, Б. А. Семенов // Вестник Воронежского ун-та. Серия: Системный анализ и информационные технологии. – 2006. – № 1. – С. 102–106.

2. *Каширина И. Л.* Генетический алгоритм решения квадратичной задачи о назначениях специального вида / И. Л. Каширина // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2003. – № 1. – С. 128–131.

3. *Каширина И. Л.* Эволюционное моделирование: учебное пособие / И. Л. Каширина. – Воронеж : ИПЦ ВГУ, 2011. – 60 с.

4. Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных заведениях. Приложение №3. (СанПиН 2.4.2.2821-10)

5. Система моделей и методов планирования и организации учебного процесса в ВУЗе // Под ред. В. В. Гусева, Н. Я. Краснера. – Изд-во Воронежский университет, 1984. – 290 с.

6. *Астахова И. Ф.* Составление расписания учебных занятий на основе генетического алгоритма / И. Ф. Астахова, А. М. Фирас // Вестник Воронежского ун-та. Серия: Системный анализ и информационные технологии. – 2013. – № 2. – С. 93–99.

7. *Пайкерс В. Г.* Методика составления расписания в образовательном учреждении / В. Г. Пайкерс. – М. : АРКТИ, 2001. – 112 с.

8. *Черноморов Г. А.* Теория принятия решений: Учебное пособие / Г. А. Черноморов. – Южно-Российский гос. техн. ун-т. Новочеркасск : Редакция журнала «Известия вузов. Электромеханика», 2002 – 276 с.

Каширина Ирина Леонидовна – доктор технических наук, доцент кафедры математических методов исследования операций факультета ПММ Воронежского государственного университета.
Тел.: 8-903-653-92-93
E-mail: kash.irina@mail.ru

Kashirina I. L. – doctor of technical Sciences, associate Professor, Voronezh State University
Tel.: 8-903-653-92-93
E-mail: kash.irina@mail.ru

Ухин Александр Леонидович – аспирант Воронежского государственного архитектурно-строительного университета.
Тел.: 8-920-417-12-19
E-mail: ukhin81@mail.ru

Ukhin A. L. – graduate student, Voronezh State University of Architecture and Civil Engineering
Tel.: 8-920-417-12-19
E-mail: ukhin81@mail.ru