

УДК 519.1:519.8:004.6

**КОНКРЕТИЗАЦИЯ ПОСТАНОВОЧНЫХ ПРИНЦИПОВ ПРОБЛЕМЫ
МНОГОКРИТЕРИАЛЬНОЙ ОПТИМИЗАЦИИ РАСПИСАНИЯ ЗАНЯТИЙ В УНИВЕРСИТЕТЕ****М.В. ДЕКАНОВА***(Полоцкий государственный университет)*

Проанализированы существующие подходы к решению задачи составления расписания, такие как: локально-эволюционный метод, многоагентный подход, интеллектуальный метод, метод замещения, метод последовательного анализа вариантов, генетический алгоритм, метод раскраски графов. Выделены достоинства и недостатки каждого из них. Сделан вывод, что задача составления расписаний университета относится к классу дискретных задач оптимизации с конечным множеством альтернатив. Решение задачи составления расписания осложняется многокритериальностью и многовариантностью. Показано, что для разрешимости рассматриваемой задачи необходимо применить комплексный подход, учитывающий достоинства всех рассмотренных в рамках данной работы методов.

Введение. Задача составления расписаний является предметом научных исследований с середины прошлого века. Область ее применения включает в себя различные сферы человеческой деятельности: транспортные перевозки, массовое обслуживание, промышленность, образование и т.д. Практика выдвигает множество задач, которые невозможно эффективно решить путем полного перебора. Для большинства моделей теории расписаний нахождение оптимального расписания является трудноразрешимой задачей, а решение приближенных к реальным условиям задач обладает ещё большей сложностью, так как данные решения должны удовлетворять многочисленным, зачастую конфликтующим между собой ограничениям производственного, организационного и психофизиологического характера.

Количественный и качественный рост высшей школы требует нового подхода к решению задач управления учебной, научной и хозяйственной деятельностью университетов. Этот подход в последние годы находит свое воплощение в применении современных средств вычислительной техники и математических методов в управлении высшими учебными заведениями. В современном мире всё большее распространение получают различного рода системы автоматизации технических процессов, которые всегда выполнялись вручную. К таким же процессам относится и составление расписания, которое до сих пор во многих учебных заведениях создается вручную на основе многолетнего опыта. Современная же наука располагает средствами, позволяющими наилучшим образом организовать любой процесс, в том числе и учебный [1].

Задача планирования расписания учебных занятий – это задача на составление расписания комбинаторного типа, характерной особенностью которой является огромная размерность и наличие большого числа ограничений сложной формы. Фактически в настоящее время не существует универсальных методов решения таких задач. Математическая (классическая) теория расписаний [1–3] охватывает лишь узкий круг хорошо формализуемых проблем, которые обычно сводятся к следующим задачам: коммивояжера, транспортной и т.п. Прямое применение данных методов к задаче составления расписания учебных занятий не представляется возможным. Тем не менее есть ряд эвристических и переборных методов, которые вполне поддаются программированию.

Изучение опыта создания подобных систем показывает, что в последние годы предпринимались множественные попытки совершенствования планирования учебного процесса путем построения алгоритмов оптимизации задач планирования учебной работы университета и последующей их реализацией на вычислительной технике. Такие исследования в разное время проводились и продолжаются в некоторых университетах [4, 5].

В данной работе рассмотрим существующие подходы к составлению расписания учебных занятий университета.

Локально-эволюционный метод

Метод решения задач дискретной оптимизации является результатом синтеза эвристического и эволюционного подходов к решению задач данного класса. Класс задач дискретной оптимизации включает в себя такие задачи, как составление оптимального расписания, маршрутная оптимизация, задача оптимального планирования многопрофильного производства и т.п.

Эвристический метод – достаточно эффективный метод поиска решений задач дискретной оптимизации большой размерности. Из него было взято на вооружение использование локальных эвристик (локальных правил поиска решения). Из *эволюционного* – параллельное развитие множества поисковых процессов и модели отбора.

Особенностью локально-эволюционного подхода является возможность осуществлять поиск в сильно ограниченном пространстве поиска. Эта особенность позволяет существенно сократить временные затраты по сравнению с классическими подходами и повысить качество самого решения.

Дополнительное сжатие пространства поиска становится возможным за счет использования при построении расписания эвристик особого типа, а именно рекурсивных.

При решении задачи используются следующие исходные данные:

- 1) $A = \{a_i\}$ – аудиторный фонд (множество аудиторий);
- 2) $T = \{t_i\}$ – плановый фонд времени (1-я, 2-я, 3-я, 4-я недели);
- 3) $W = \{w_i\}$ – множество преподавателей;
- 4) $P = \{p_i\}$ – множество плановых занятий;
- 5) $G = \{g_i\}$ – множество групп учащихся;
- 6) $U = \{<p_i, g_i>\} \subset P \times G$ – множество учебных планов для групп учащихся (закрепляет за каждой группой учащихся определенный объем плановых занятий).

Недопустимым планом расписания является план, содержащий хотя бы один конфликт по преподавателю или группе, или аудитории, поэтому при решении задачи данным методом используются следующие ограничения:

- 1) $A: a_1 \neq a_2$;
- 2) $W: w_1 \neq w_2$;
- 3) $U/\{p_1\} \cap U/\{p_2\} = \emptyset$.

Для занятий, длительность проведения которых больше одного кванта (два академических часа или «пара») планового времени, вводится *дополнительное ограничение* последовательного по времени захвата квантового времени.

В расписании представлены все плановые занятия из P – это *требование* полноты расписания.

Обычно перед составлением расписания закрепляют сочетания (w, p) в виде нагрузки преподавателя, благодаря чему сочетание (w, p) в расписании присутствует и в нагрузке.

В задаче составления оптимального расписания нужно найти расписание с минимальной ценой. Поэтому в данном методе решения задачи используется в качестве оценочной функции следующая эвристическая функция:

$$Q(S) = (a_1G_1 + a_2G_2 + a_3G_3 + a_4G_4 + a_5G_5) + (a_5A_1 + a_6A_2) + (a_7W_1 + a_8W_2 + a_9W_3) + a_{10}P_1, \quad (1)$$

где G_1 – количество межкорпусных переходов в течение дня у групп (подгрупп) учащихся; G_2 – количество «форточек» в расписании групп (подгрупп); G_3 – степень разнообразия занятий в течение дня у групп (подгрупп); G_4 – степень отклонения от вида планирования занятий; G_5 – выход за пределы рекомендуемого календаря учебного времени; A_1 – количество «форточек» в расписании аудитории; A_2 – степень загрузки аудиторий во время занятий; W_1 – количество межкорпусных переходов в течение дня у преподавателя; W_2 – выход за пределы рекомендуемого календаря рабочего времени преподавателя; W_3 – количество «форточек» в течение дня у преподавателя; P_1 – степень покрытия неполным расписанием учебных планов; $a_1, a_2, a_3, \dots, a_{10}$ – масштабирующие коэффициенты.

Выбор масштабирующих коэффициентов отражает принятую оценку понятия «хорошее» расписание. Но поскольку это представление субъективно, то возникает проблема противоречивости. Осуществляется поиск компромисса между противоречащими факторами. Система правил едина, поэтому невозможно автоматически сделать идеальное расписание одному преподавателю, проигнорировав интересы других, и расписание, рассматриваемое одним преподавателем как «хорошее», будет не соответствовать представлениям о «хорошем» расписании другого преподавателя или студента.

В локально-эволюционном алгоритме разработаны две рекурсивные эвристики для рассматриваемой задачи:

- 1) HSG – планирование занятий;
- 2) HSX – перенос занятий в расписании.

Первая эвристика предназначена для выращивания расписания и способна работать с неполным расписанием; вторая – для оптимизации полного расписания.

HSG: действие данной эвристики состоит из двух шагов. Во-первых, выбирается занятие, которого еще нет в расписании. Во-вторых, выбирается время и аудитория для проведения данного занятия. При реализации вставки работают эвристики, поощряющие «хорошее» расписание и штрафующие за «плохое», выигрывает вариант, обеспечивающий максимум оценочной функции.

HSX: перенос занятия p_{a4} в плане $s_a = (a_{a1}, t_{a2}, w_{a3}, p_{a4})$ в другую аудиторию и в другое время. То есть изменение значения сочетания (a_{a1}, t_{a2}) на (a_{b1}, t_{b2}) . Если при этом существует $s_b = (a_{b1}, t_{b2}, w_{b3}, p_{b4})$, то данная процедура повторяется для p_{b4} . С целью устранения заикливания глубина рекурсии должна быть искусственно ограничена.

Достоинство метода – получение хороших приближенных решений уже на первых итерациях и высокая вероятность получения точного решения за достаточно большое число итераций. Очевидное преимущество локально-эволюционного алгоритма – снижение вычислительной сложности алгоритма по сравнению с классическими алгоритмами. Еще одним преимуществом является гибкость, которая выра-

жается в совместном действии различных стратегий, а также возможность видоизменять участвующие в нем эвристические стратегии.

Важная особенность локально-эволюционного алгоритма – недетерминистический характер вычислений, который является следствием внутреннего параллелизма вычислений. Это выражается в асинхронности действия различных локальных стратегий.

Основной недостаток алгоритма – его эвристический характер. Сочетание различных стратегий позволяет преодолевать большее число локальных минимумов по сравнению с обычным эвристическим алгоритмом. Еще один очевидный недостаток – располагать можно только лишь статистическими оценками эффективности данного алгоритма, что требует проведения огромного количества экспериментов на различных задачах для получения достоверной информации [6].

Многоагентный подход

В информатике и программной инженерии агент – самостоятельный объект системы, обладающий свойством коммуникабельности и наделённый собственной системой принятия решений.

Коммуникабельность – способность агента обмениваться сообщениями с другими агентами и прочими объектами системы.

Сутью применения агентного подхода к какой-либо задаче является то, что задача разбивается на несколько более мелких задач. Для решения каждой мелкой задачи выделяется агент. Цель агента – найти такое решение своей задачи, чтобы оно согласовалось с решениями других агентов. Агенты добиваются согласования друг с другом путем обмена информационными сообщениями. В итоге агенты находят решения для своих задач, а значит и для исходной задачи; либо выясняется, что решения нет. Идея применения агентного подхода к задаче составления расписания состоит в том, чтобы заменить агентами каждого пользователя расписания. Цель каждого агента – найти такое расписание для пользователя, чтобы оно соответствовало ограничениям задачи и ограничениям самого пользователя, имело бы наибольший уровень пользы для пользователя и не противоречило расписаниям других пользователей.

Все агенты запускаются в рамках некоторого контейнера (агентной системы). Агенты выполняют локальные вычисления, обмениваются сообщениями, что в итоге приводит к решению задачи, то есть построению расписания занятий. В идеале пользователь расписания может один раз запрограммировать своего агента-представителя, который будет замещать его в дальнейшем по всем вопросам составления расписания проведения занятий и встреч, общаясь напрямую с агентами других агентов. Кроме того, отпадает необходимость в централизованном сборе и хранении предпочтений и пожеланий пользователей расписания, так как все эти данные хранит агент, другим агентам достаточно обратиться к нему с запросом на получение необходимых данных. Пользователь может в любой момент перепрограммировать своего агента и через некоторое время увидеть результат – новое расписание, соответствующее его новым требованиям.

Основа построения собственного многоагентного алгоритма составления учебного расписания – точная постановка задачи в виде математической модели. В целях упрощения изложения в такой модели будем называть лиц, заинтересованных в результатах процесса составления расписания, пользователями расписания. К ним относятся преподаватели и учебные группы. Роль первых – провести лекционное или практическое занятие; роль последних – присутствовать на занятиях. Введём следующие обозначения.

Учебные группы и потоки: $g \in G$ – номер учебной группы; G – множество номеров всех учебных групп; $|G| = \gamma$ – количество групп. Каждая группа входит в один или несколько потоков. При объединении групп в один поток используются следующие принципы:

- 1) группы в потоке используют один и тот же аудиторный фонд;
- 2) лекции читаются всему потоку одновременно;
- 3) у каждого потока есть хотя бы одно занятие;
- 4) поток может состоять из одной учебной группы.

Пусть R – множество номеров учебных потоков; $|R| = \rho$ – количество потоков; $r \in R$ – номер потока. Каждая группа самостоятельный поток, поэтому $\rho \geq \gamma$.

Допустим, что $C_r \subset G$ – поток. Поток называется любое подмножество множества групп. $C = \{C_1, C_2, \dots, C_p\}$ – множество всех потоков. Потоки могут пересекаться между собой.

Преподаватели: $p \in P$ – номер преподавателя; P – множество всех преподавателей. $|P| = \pi$ – количество преподавателей.

Пользователи расписания: объединение множества всех групп с множеством всех преподавателей: $M = G \cup P$, $t \in M$ – уникальный номер пользователя расписания.

Время: $w \in W_g$ – номер учебного дня недели; $W_g \in W = \{1, 2, \dots, 7\}$ – множество учебных дней группы; g, W – множество всех дней недели; $j \in J = \{1, 2, \dots, 8\}$ – номер учебной пары; $T = \{(w, j) | w \in W; j \in J\}$ – множество таймслотов. Таймслотом называется элементарная единица времени в задаче составления расписания. Например, таймслот (1, 2) означает: понедельник, вторая пара. Для каждого пользователя t известно множество таймслотов, когда он свободен, т.е. в это время занятие может быть проведено $T_m^+ \subset T$,

и множество недоступных таймслотов $T_m^- \subset T(T_m^+ \cup T_m^- = T; T_m^+ \cap T_m^- = \emptyset)$, например, выходные дни, когда занятия не могут быть проведены.

Занятия: преподаватели проводят лекционные и практические занятия; лекционные занятия проводятся у всего потока сразу; практические занятия – только у одной группы; требуют другого аудиторного фонда. Например, практические занятия по информатике должны проходить в компьютерном классе. Используются следующие обозначения для занятий: $S_r = \{1, 2, \dots, \sigma_r\}$ – множество номеров лекционных занятий, читаемых на потоке r ; $s_r \in S_r$ – номер лекционного занятия; $Q_r = \{1, 2, \dots, \theta_r\}$ – множество номеров практических занятий, проводимых на потоке r ; $q_r \in Q_r$ – номер практического занятия.

Глобально уникальным идентификатором занятия будет пара «номер потока – номер занятия на потоке». Занятия существенно зависят от потока. Лекционное занятие однозначно идентифицируется парой значений «номер потока – номер лекционного занятия»: (r, s_p) .

$$RS = \{(r, s_r) | r \in R, s_r \in S, \quad (2)$$

где RS – множество всех лекционных занятий.

Количество лекционных занятий

$$|RS| = \sum_{r=1}^p \sigma_r. \quad (3)$$

Практическое занятие однозначно идентифицируется тройкой значений «номер потока – номер практического занятия – номер группы»: $(r, g, q_r) \in RQG$.

$$RQG = \{(r, g_r, q_r) | r \in R, s_r \in C_T\}, \quad (4)$$

где RQG – множество всех практических занятий.

Количество практических занятий:

$$|RQG| = \sum_{r=1}^p |C_r| \cdot \theta_r, \quad (5)$$

где $|C_r|$ – количество групп в потоке.

При дальнейшем описании задачи нам в большинстве случаев будет неважно, какое занятие рассматривается – лекционное или практическое. Поэтому под занятием будем понимать элемент из объединения двух множеств – множества лекционных и множества практических занятий:

$$E = RS \cup RQG. \quad (6)$$

Учебный план закрепляет за каждым преподавателем предметы, которые он должен будет провести в течение семестра. Назначение лекционных занятий:

$$\delta_1 : RS \rightarrow P, \quad (7)$$

где P – множество преподавателей; RS – множество лекционных занятий.

В общем случае учебный план можно записать как $\delta_1 : E \rightarrow P$. Отсюда следует:

$$\begin{aligned} \delta_1(e) &= \{\delta_1(e), e \in RS, \\ \delta_2(e) &= \{\delta_1(e), e \in RS. \end{aligned} \quad (8)$$

Зная учебный план, можно вычислить величину E_m – множество занятий, проводимых пользователем-преподавателем (или для пользователя-группы) с номером m :

$$E_m = \{e | m \in P \wedge \delta(e) = m\} \cup \{e = (r, s) | m \in C_r \wedge s \in S_r\} \cup \{e = (r, q, s) | m \in C_r \wedge q \in Q_r\}. \quad (9)$$

Аудиторный фонд – множество аудиторий, где могут быть проведены занятия; A – множество всех аудиторий и помещений. Для каждого занятия e выделяется некоторое подмножество аудиторий $A_e \subset A$; аудитория для проведения занятия выбирается только из этого подмножества.

Неизвестные функции задачи: неизвестные в данной задаче – расписание времени и расписание аудиторий; расписание времени – отображение множества предметов на множество таймслотов:

$$\tau : E \rightarrow T, \quad (10)$$

где E – множество всех занятий; T – множество всех таймслотов.

Расписание аудиторий – это отображение множества занятий на множество аудиторий:

$$\alpha: E \rightarrow A, \quad (11)$$

где E – множество всех занятий; A – множество всех аудиторий.

В задаче используются следующие ограничения.

1) один преподаватель в каждый момент времени может проводить не более одного занятия:

$$\forall p \in P, \forall e_1, e_2 \in E: e_1 \neq e_2 \wedge \delta(e_1) = \delta(e_2) = p \Rightarrow \tau(e_1) \neq \tau(e_2); \quad (12)$$

2) в одной аудитории в каждый момент времени может проводиться не более одного занятия:

$$\forall a \in A, \forall e_1, e_2 \in E: e_1 \neq e_2 \wedge \alpha(e_1) = \alpha(e_2) = a \Rightarrow \tau(e_1) \neq \tau(e_2); \quad (13)$$

3) у одной группы в каждый момент времени может проводиться не более одного занятия:

$$\begin{aligned} \forall g \in G: (e_1 = (r_1, \dots), e_2 = (r_2, \dots)) \in E \wedge g \in C_{r_1} \wedge g \in C_{r_2} \wedge e_1 \neq e_2 \vee; \\ \vee (e_1 = (\dots, g), e_2 = (\dots, g)) \in E \wedge e_1 \neq e_2 \Rightarrow \tau(e_1) \neq \tau(e_2). \end{aligned} \quad (14)$$

Другими словами, для каждого двух пересекающихся потоков в каждый момент времени не может быть разных лекционных занятий в одно и то же время, и для каждой группы не может быть разных практических занятий в одно и то же время. Это ограничение удобно проверять, предварительно разбив все занятия на γ классов. В класс входят все занятия, которые читаются группе g персонально или в каком-либо потоке:

$$K_g \subset E, g \in G (g_1 \neq g_2 \Rightarrow K_{g_1} \cap K_{g_2} = \emptyset). \quad (15)$$

Ограничение 3 будет выполнено тогда и только тогда, когда все занятия групп одного класса проводятся в разное время.

Опишем *приоритеты занятий*. Очевидно, что не все занятия имеют одинаковую важность в рамках учебного процесса. Для них можно задать порядок по приоритету. Более приоритетные занятия будут в первую очередь занимать время и место; менее приоритетные занятия уже не смогут претендовать на эти ресурсы. Итак, приоритеты занятий – отношение частичного порядка на множестве занятий:

$$e_1 > e_2 \Leftrightarrow U(e_1) \geq U(e_2), \quad (16)$$

где $U(e)$ – «полезность» занятия e :

$$(U(e) = |M_e| + k_1(e) + k_2(e) + k_3(p_e);$$

M_e – множество групп, у которых проводится занятие e :

$$M_e = \{m | (e = (r, s)) \in RS \wedge m \in C_r) \vee (e = (r, q, m)) \in RQG\};$$

$k_1(e) \in \{0, 5, 10\}$ – характеризует степень важности занятия для специальности (0 – «ненужные занятия»; 5 – важное, но не по специальности; 10 – занятие по специальности); $k_2(e) \in \{0, 2\}$ – 0 – младшие курсы; 2 – старшие курсы; $p_e = p$, $\delta(e) = p$ – номер преподавателя, который ведет занятие e ; $k_3(p_e) \in \{0 \dots 5\}$ – оценка преподавателя как научного сотрудника: больше значит, что преподаватель более ценный научный сотрудник; в случае равенства занятия упорядочиваются в лексикографическом порядке.

Предпочтения пользователей. Важная часть предлагаемой модели – предпочтения пользователей. Каждое предпочтение – это числовая оценка в диапазоне от 0 до 1; 0 соответствует наименьшему уровню предпочтения; 1 – наибольшему. В модели имеют место два вида предпочтений: предпочтения пользователя $m \in M$ о времени проведения занятия; предпочтения пользователя $m \in M$ о месте проведения занятия.

Качество решения ограничивается следующими критериями:

- критерий качества – обобщенный критерий, состоящий из нескольких частных критериев – оценивает найденное решение, т.е. пару функций;

- частный критерий $F_e^1(\tau) = \sum_{m \in M_e} f_m^1(e, \tau(e)) \rightarrow \max \sum_{r=1}^p \sigma_r$ определяет сумму предпочтений пользователей о времени занятия e ;

- частный критерий $F_e^2(\alpha) = \sum_{m \in M_e} f_m^2(e, \alpha(e)) \rightarrow \max$ определяет сумму предпочтений пользователей о месте проведения занятия e ;

- обобщенный критерий $F(\tau, \alpha) = \sum_{e \in E} F_e^1 + F_e^2 \rightarrow \max$ в начале максимизирует пару частных критериев для наиболее приоритетного занятия, затем для следующего по приоритету и т.д. В паре критериев вна-

чале максимизируется первый критерий (время), затем второй (место). Такая структура гарантирует, что приоритетные занятия получают лучшее время и место. Считается, что время проведения занятия важнее при составлении расписания, поэтому вначале определяется время, а затем место проведения занятия.

Решением задачи являются неизвестные функции при условии выполнения всех ограничений и максимальном значении критерия качества.

Основные достоинства агентного подхода:

1) сильная индивидуальность предпочтений (каждый пользователь имеет возможность высказать свои предпочтения, заложить любую программу принятия решений в своего агента);

2) отсутствие предварительного сбора заявок (в общем случае агенты могут без ограничений подключаться к процессу решения задачи, предварительный сбор информации о предпочтениях каждого агента не требуется);

3) динамическое изменение предпочтений (каждый агент может поменять свои предпочтения в любой момент времени, это приведет к обновлению расписания).

4) существование частичного решения (в каждый момент времени существует частичное решение задачи).

К недостаткам агентного подхода можно отнести:

1) распределенность (агент – это самостоятельный программный модуль, все агенты исполняются на разных компьютерах);

2) резко (экспоненциально) увеличиваются временные затраты на поиск лучшего (приемлемого) решения с ростом размерности решаемой задачи;

3) большой риск (отсутствует гарантия получения приемлемого решения задачи построения расписания занятий);

4) практически невозможным (в силу большой размерности, громоздкости и сложности математической модели решаемой задачи) становится оценивание влияния на решение задачи ограничений, накладываемых на алгоритм, оценивание критичности полученного решения к данным ограничениям.

Интеллектуальный метод

В основе данного метода лежит использование различного рода эвристик или эвристических алгоритмов, при разработке которых используются интуитивные предположения, не подкрепленные соответствующим математическим обоснованием. Формирование расписания занятий с помощью некоторых правил (эвристик) позволяет несколько ускорить поиск наилучшего расписания, но использование таких алгоритмов в большинстве случаев гарантирует лишь нахождение приближенного решения (достижение локального экстремума). В этом случае возникает проблема оценки близости найденного локального экстремума к глобальному экстремуму.

Эта проблема решается путем сравнения расписания, полученного эвристическим методом и расписания, полученного методом перебора для близкой задачи малой размерности. Несмотря на указанный недостаток, эвристические алгоритмы продолжают оставаться достаточно эффективным инструментом поиска «лучшего» в некотором смысле решения в тех случаях, когда нахождение наилучшего решения крайне затруднено или невозможно.

При составлении расписания занятий в образовательных системах массового обучения предлагается использовать математический аппарат нечеткой логики для перехода от «жестких» ограничений к более «мягким» требованиям.

Однако несмотря на то, что данный подход позволяет значительно упростить формализацию требований, предъявляемых к расписанию, рассмотрение наиболее важных требований в качестве «мягких» требований может привести к «плохому» расписанию (появлению «окон», «накладок» и т.д.).

Метод замещения

Расписание учебных занятий для учебных заведений (школ, колледжей, вузов и т.п.) относится к классу задач целочисленного программирования с булевыми переменными. Французские специалисты по методам исследования операций А. Кофман и А. Анри-Лабордер называют ее задачей «планирования загрузки преподавательского состава и обеспечения студентов аудиториями».

Формально задачу составления учебного расписания можно поставить следующим образом:

Задано множество преподавателей T , множество «учебных дисциплин» (учебный план) D , множество аудиторий H и множество рабочих часов (часовая сетка) S .

Требуется найти расписание, обслуживающее все дисциплины, удовлетворяющее требованиям преподавателей к проведению занятий и не имеющее противоречий и накладок относительно аудиторий.

Для формирования допустимого расписания в работе рассматривается возможность применения метода замещения (МЗ), который является вариантом метода ветвей и границ с использованием двудольного графа со взвешенными ребрами и предписанными степенями вершин. В качестве ограничений, исходящих от преподавателей и условий конкретного учебного заведения, предлагается использовать так называемые вектора топологии.

Решение задачи составления расписания сводится к следующей схеме:

- множество преподавателей T и множество учебных дисциплин D образуют соответственно две доли графа;
- матрица весов формируется на основе начальных условий (а именно, множеств D и S).

Требуется найти каркас, удовлетворяющий целевой функции $\sum_{i=1}^T \sum_{j=1}^D c_{ij} x_{ij} \rightarrow \max$, где $x \in \{0,1\}$, c_{ij} – вес ребра, соединяющего вершины i и j , $c_{ij} \sum_{i=1}^T \sum_{j=1}^D x_{ij} = m$, где m – число ребер в исходном подграфе, и ограничениям $A_{\max} = (A_1, \dots, A_i, \dots, A_n)$, $A_{\min} = (a_1, \dots, a_i, \dots, a_n)$, где $A_{\max}$, $A_{\min}$ – векторы допустимых степеней (так называемые векторы топологии).

Алгоритм сводится к следующим построениям.

1) После упорядочения списка ребер исходного графа по возрастанию весов по первым $n - 1$ ребрам вычисляется вектор начальных степеней $V^0 = v_1^0, \dots, v_i^0, \dots, v_j^0$ и номер вершины с максимальным дефицитом степени $R = \max_{i=1} \{S_i - v_i^0\}$, где S – вектор закрепленных степеней. Просматривая по возрастанию нумерации список ребер, находим первое ребро, инцидентное вершине с максимальным значением дефицита. Ребро помечается;

2) выполняя $n - 3$ итерации, аналогичных пункта 1, находим следующие ребра, инцидентные изолированным вершинам наращиваемого начального подграфа;

3) к полученным ребрам добавляется ребро из числа непомеченных, которое помечается. Из помеченных ребер формируется начальный подграф путем упорядочения его множества ребер E^0 по возрастанию весов. Непомеченные ребра упорядочиваются также по возрастанию весов и становятся множеством E/E . Пометки с множества ребер E^0 снимаются. Дальнейшее решение задачи выполняется по известной схеме замещений.

В результате получаем граф, ребра которого образуют соответствие между множеством вершин первой доли графа T и множеством вершин второй доли графа D .

Преимущество алгоритма, сформированного на основе метода замещений по сравнению с алгоритмами, реализующими метод ветвей и границ, заключается в том, что алгоритмы метода замещений позволяют следить за топологией искомого подграфа даже при существенных отличиях в степенях его вершин при решении задач со списком степеней, также позволяют формировать правило остановки алгоритма по функциям от количества исследованных узлов в дереве решений, от таймера ЭВМ и т.д., устраняя «проклятие размерности» при дефиците вычислительных ресурсов путем выдачи приближенного решения.

Метод последовательного анализа вариантов

На основе обобщения идей теории последовательных решений и динамического программирования была разработана общая схема последовательного анализа вариантов (ПАВ). С точки зрения формальной логики схема ПАВ сводится к такой последовательности повторения процедур:

- 1) разбиение множества вариантов решения задачи на несколько подмножеств, каждое из которых обладает специфическими свойствами;
- 2) использование этих свойств для поиска логических противоречий в описании отдельных подмножеств;
- 3) исключение из дальнейшего рассмотрения тех подмножеств вариантов решения, в описании которых имеются логические противоречия.

Методика последовательного развития, анализа и отсева вариантов состоит в таком способе развития вариантов и построения операторов их анализа, которые позволяют отсеивать бесперспективные начальные части вариантов до их полного построения.

Поскольку при отсеивании бесперспективных начальных частей вариантов отсеивается тем самым и все множество их продолжений, происходит значительная экономия вычислительных затрат.

На базе этой общей схемы был разработан целый ряд алгоритмов последовательного анализа вариантов, которые получили широкое применение в практике.

Приближенные методы можно разбить на следующие группы: методы локальной оптимизации, модификации точных методов; эвристические методы, максимально учитывающие специфику решаемых задач; методы случайного поиска; методы, сочетающие локальную оптимизацию со случайным поиском.

Отметим, что многие приближенные алгоритмы позволяют решать задачи дискретной оптимизации в диалоговом режиме. Это дает возможность в зависимости от выделяемых ресурсов (времени, памяти компьютера и т.п.) последовательно улучшать полученное решение путем изменения всех или некоторых исходных данных.

Значительная часть приближенных алгоритмов дискретной оптимизации базируется на использовании вычислительных схем известных точных методов, таких как метод ветвей и границ и метод последовательного анализа вариантов и прочих.

Одними из наиболее развитых приближенных методов являются методы локальной оптимизации, имеющие своей целью отыскание локально оптимальных решений. Нередко в этих методах на определенных этапах решения задачи используются методы случайного поиска, а также различные способы (эвристики), которые дают возможность сократить ход вариантов и максимально учитывают специфику задачи.

Следует отметить, что алгоритмы, в которых комбинируются различные идеи, на практике часто оказываются наиболее эффективными. С помощью этих методов были решены многочисленные сложные задачи классификации объектов, размещения, планирования и проектирования. Основное преимущество этих методов – простота реализации, а основной недостаток – это то, что они не могут адаптироваться к условиям решаемой задачи. Значительно более гибкими являются методы, у которых вероятностный закон зависит от исходов предыдущих испытаний и изменяется от итерации к итерации. Это методы случайного поиска с обучением.

Методы случайного поиска используются для приближенного решения многомерных задач о ранге, а также задач линейного булева программирования большой размерности.

Детализация схемы ПАВ происходила в нескольких направлениях. В связи с решением задач линейной или древовидной структуры был сформирован обобщенный принцип оптимальности для монотонно рекурсивных функционалов, на основе которого можно строить схему решений, свободную от некоторых ограничений, присущих каноническим процедурам динамического программирования.

Для структур, которые имеют форму дерева, были исследованы вопросы, связанные с порядком просмотра ветвей, и найдены способы минимизации числа массивов информации, необходимой для реализации процедур ПАВ.

Метод ПАВ использовался для решения большого количества важных задач оптимального планирования и проектирования, таких как: расчет транспортных сетей; размещение на сети типа дерева; проектирование распределительных электрических сетей; выбор оптимальных параметров магистральных газопроводов и т.п.

Далее на основе алгоритмической схемы ПАВ были разработаны эффективные методы конструирования, анализа и отсева вариантов для решения многочисленных задач календарного планирования. Относительно теории календарного планирования (КП) были обоснованы точные методы решения разных классов задач малой размерности, доказаны необходимые и достаточные условия приоритетной решаемости для задач КП с однотипным оборудованием, разработаны эффективные способы, которые используют правила доминирования.

Для решения задач большой размерности дискретного сепарабельного и линейного целочисленного программирования были разработаны декомпозиционные алгоритмы, которые базируются на схемах последовательного анализа и отсева вариантов. При этом сужение исходного множества вариантов осуществлялось путем отсева отдельных компонент, что дало возможность свести решение исходной задачи большой размерности к решению совокупности подзадач малой размерности.

Метод ПАВ базируется на отсеке бесперспективных элементов как по ограничениям, так и по целевой функции. Его идеи рассмотрим на примере задачи дискретного программирования.

Схему метода ПАВ изложим для следующей задачи:

$$\min_{x \in D(X)} f(x).$$

При ограничениях:

$$\begin{aligned} q_i(x_1, \dots, x_n) &\leq 0, i = \overline{1, m}, \\ x_j &= Q_j, j = \overline{1, n}, \end{aligned} \quad (17)$$

где $Q_j = (q_{1j}, \dots, q_{mj})$ – заданные конечные множества. Вектор $x = (x_1, \dots, x_n)$ назовем *решением*, если его компоненты $x_j \in Q_j, j = \overline{1, n}$.

Множество всех решений обозначим через Ω . Решение называется *допустимым*, если оно удовлетворяет неравенству (17).

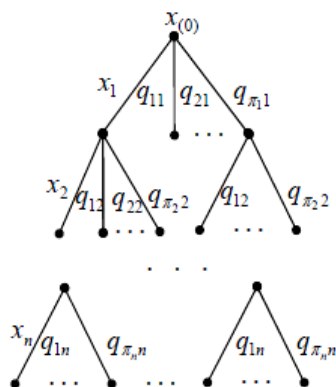
Множество всех допустимых решений обозначаем через Ω_j .

Вектор $x_{(p)}(x_1, \dots, x_n)$, $p < n$ будем называть *частичным решением*, если $x_j \in Q_j$.

Если при этом он может быть достроен до допустимого решения $(x_1, \dots, x_p, x_{p+1}, \dots, x_n)$, будем называть его *допустимым частичным решением*.

Множество всех частичных решений задачи схематически можно изобразить в виде дерева решений H , изображенного на рисунке. Вершинам его взаимно-однозначно соответствуют частичные решения, причем висячим вершинам соответствуют полные решения $p \in n$. Процесс решения задачи с помо-

щью метода ПАВ можно интерпретировать как продвижение на дереве H , начиная с его корня, отвечающего частичному решению $x_{(0)}$.



Дерево решений

Метод эффективен для задач небольшой размерности, так как при большой размерности задачи время решения задачи будет существенно увеличиваться.

Генетический алгоритм

Генетический алгоритм – новейший, но не единственно возможный способ решения задач оптимизации. С давних пор известно два основных класса методов решения таких задач – переборный и эвристический. Идея целесообразности использования генетического алгоритма состоит в том, что он, комбинируя переборные и эвристические методы, позволяет находить приближенное решение, точность которого возрастает при увеличении времени расчета.

Известно, что генетический алгоритм зарекомендовал себя как быстрый метод, позволяющий получать приемлемые решения за допустимое время, которому присущи как преимущества, так и недостатки.

Основные объекты генетических алгоритмов – это особь (индивидуум), хромосома, ген, генотип, фенотип, функция пригодности.

Основными операциями, или, как это принято в теории генетических алгоритмов, операторами, выполняемыми над объектами, являются селекция, скрещивание и мутация. Особь включает в себя одну или несколько хромосом, каждая хромосома в свою очередь состоит из одного или нескольких генов.

В качестве логической модели особи выступает кортеж кодов хромосом. Генотипом особи принято считать упорядоченный набор генов некоторой особи, фенотипом особи – набор значений (признаков) этих генов. Допустима такая аналогия: генотип – упорядоченная совокупность варьируемых переменных; фенотип – упорядоченная совокупность значений этих переменных; функция пригодности – целевая функция оптимизационной задачи целочисленного программирования.

Предлагается использовать в качестве генов настроечные коэффициенты w_j алгоритма составления расписания. В результате мутаций будут получены новые наборы настроечных коэффициентов, а, как следствие, результаты работы алгоритма будут различны. При этом оценку качества составленного расписания следует проводить с использованием значений w_j заданных диспетчером. Таким образом, учитывается возможность генерации расписания, наиболее удовлетворяющего оценкам качества с точки зрения диспетчера, при использовании несколько измененных оценок качества в процессе его генерации. Хромосома такого генетического алгоритма будет представлять собой набор действительных чисел: $\{w_1, w_2, w_3, \dots, w_m\}$, где $w_1 \dots w_m$ – гены хромосомы, настроечные коэффициенты алгоритма составления расписания; m – количество настроечных коэффициентов алгоритма.

Целью работы генетического алгоритма является достижение максимума функционала:

$$R_{il} = \sum_{j=1}^m w_j k_{jl},$$

$$R_i = \max(R_{il}), \quad l = 1, \dots, h,$$

$$R = \sum_{i=1}^h R_i \rightarrow \max,$$

где R_{il} – качество расположения i -го занятия на l -й позиции в расписании; k_{jl} – значение, полученное по j -му критерию оценки качества расположения занятия на l -й позиции в расписании; w_j – весовой коэф-

фициент j -го критерия оценки качества; m – количество критериев оценки качества; l – возможная позиция i -го занятия в расписании; R_i – качество расположения i -го занятия в расписании; h – количество возможных вариантов расположения занятия в расписании; R – качество составленного расписания.

Преимущества генетического алгоритма:

- 1) возможность получения близких к оптимальному решений за небольшое вычислительное время;
- 2) результатом поиска является не одно найденное решение, а сразу несколько, что позволяет выбрать из них наиболее подходящее;
- 3) на любой итерации вычисления имеется возможность получить наилучшее на данный момент решение, что важно для задач высокой размерности;
- 4) для задач высокой размерности скорость работы алгоритма можно «регулировать» размерами начальной популяции.

Недостатки генетического алгоритма:

- 1) качество решения значительным образом зависит от размера начальной популяции, что составляет проблему использования памяти на первых итерациях;
- 2) решения, полученные в результате нескольких экспериментов для одной и той же задачи, могут незначительно различаться;
- 3) недостаточное разнообразие хромосом в популяции может привести к преждевременному окончанию работы алгоритма и, как следствие, к получению некорректного расписания. Далее, завершение работы генетического алгоритма происходит по достижению заданного (не всегда обоснованного) числа итераций, что в ряде случаев препятствует поиску лучшего расписания;
- 4) слабый учет специфики задачи составления расписания учебных занятий при организации ее решения. Так, в известных генетических алгоритмах составления расписания занятий не учитываются наличие связей между объектами расписания и большое многообразие описаний самих объектов расписания.

Метод раскраски вершин графа

В процессе решения задачи построения расписания необходимо назначить каждое занятие на конкретный промежуток времени рабочего дня недели. Поскольку длительность одной пары фиксирована, то заранее известны все возможные интервалы времени, на которые могут быть назначены пары.

Обозначим через W множество всех таких известных интервалов времени в рассматриваемом при составлении расписания плановом периоде (например, в течение шестидневной рабочей недели).

Пронумеруем интервалы множества W , начиная с первого по времени интервала, и будем считать элементы множества W упорядоченными в соответствии с номерами его интервалов: $N = \{1, 2, \dots, n\}$, где $n = |W|$.

Если для проведения какого-либо занятия аудитория не была назначена заранее (до составления расписания), в процессе составления расписания необходимо также назначить аудиторию для проведения такой пары. Расписания занятий в вузе являются периодическими, поэтому достаточно составить расписание учебных занятий на один плановый период. Обычно периодом является одна неделя или же две последовательные недели (если расписания для них существенно отличаются одно от другого). Нарушения периодичности, возникающие в процессе реализации расписания, можно учитывать в режиме корректировки основного расписания, составленного на один плановый период. При построении расписания учебных занятий необходимо выполнить следующие требования (условия):

- 1) если два занятия проводятся с одной и той же группой студентов, то они должны быть назначены на различные интервалы времени;
- 2) если два занятия проводятся одним преподавателем, то они должны быть назначены на различные интервалы времени;
- 3) если два занятия проводятся в одной и той же аудитории, то они должны быть назначены на различные интервалы времени;
- 4) пусть V_k обозначает множество всех занятий, которые должны быть проведены в аудиториях типа k . Тогда мощность подмножества занятий множества V_k , которые могут быть назначены на один и тот же интервал времени из множества W , не должна превышать общего числа имеющихся в вузе аудиторий типа k ;
- 5) необходимо предусмотреть возможность запрещать назначение пар для конкретной группы учащихся на тот или иной интервал времени из множества W ;
- 6) необходимо предусмотреть возможность запрещать назначение пар для преподавателя на тот или иной интервал времени (например, в связи с методическими днями или в связи с личными пожеланиями преподавателя);
- 7) для студентов в построенном расписании не должно быть так называемых форточек.

Для представления исходных данных задачи построения оптимального расписания занятий, а также для представления процесса ее решения предлагается использовать сетевую модель. При этом процесс построения расписания сводится к раскраске вершин V графа $G = (V, E)$ с множеством ребер E и дополнительными ограничениями на множества цветов, которые разрешается использовать для различных вершин из множества V .

Выбор теоретико-графовой модели определяется тем, что процесс обучения в вузе можно считать детерминированным, поскольку для каждой группы учащихся заранее определен список предметов (курсов), и этот список не изменяется в образовательном процессе в течение длительного периода времени. Преподавательский коллектив и группы студентов, как правило, остаются неизменными во время всего планового периода. В силу такой детерминированности задача построения расписания может быть сформулирована в терминах раскраски вершин графа. Функцию φ называют раскраской (вершин) графа $G = (V, E)$, если для каждой вершины $v_i \in V$ она определяет натуральное число (называемое цветом) $\varphi(v_i) \in N$ так, что из включения $[v_i, v_j] \in E$ следует соотношение $\varphi(v_i) \neq \varphi(v_j)$. Здесь N обозначает подмножество множества натуральных чисел. В задаче оптимальной раскраски графа требуется построить раскраску $\varphi: V \rightarrow N$ вершин графа $G = (V, E)$, которая содержит минимальное число цветов.

Заключение

В рамках данной работы были проанализированы следующие подходы к решению задачи составления расписаний: локально-эволюционный метод, многоагентный, интеллектуальный, метод замещения, метод последовательного анализа вариантов, генетический алгоритм, метод раскраски графов.

Таким образом, можно сделать вывод, что задача составления расписаний университета относится к классу дискретных задач оптимизации с конечным множеством альтернатив.

Решение задачи составления расписания чрезвычайно осложняется многокритериальностью и многовариантностью. Для разрешимости рассматриваемой задачи необходимо применить комплексный подход, учитывающий достоинства всех рассмотренных в рамках данной статьи методов.

ЛИТЕРАТУРА

1. Лазарев, А.А. Методы и алгоритмы решения задач теории расписаний для одного и нескольких приборов и их применение для задач комбинаторной оптимизации: дис. ... д-ра физ.-мат. наук / А.А. Лазарев. – М.: Ин-т проблем управления им. В.А. Трапезникова РАН, 1997. – 413 с.
2. Баронов, В.В. Автоматизация управления предприятием / В.В. Баронов. – М.: Инфра-М, 2000.
3. Логоша, Б.А. Комплекс моделей и методов оптимизации расписания занятий в вузе / Б.А. Логоша, А.В. Петропавловская // Экономика и математические методы. – 1993. – Т. 29, № 4.
4. Танаев, В.С. Теория расписаний. Многостадийные системы / В.С. Танаев, Ю.Н. Сотсков, В.А. Струсович. – М.: Наука, 1989. – 328 с.
5. Гафаров, Е.Р. Задачи теории расписаний. Алгоритмы и применение / Е.Р. Гафаров // Современные проблемы фундаментальных и прикладных наук: тр. 49 науч. конф. МФТИ: Управление и прикладная математика. – Москва-Долгопрудный, 2006. – С. 82–83.
6. Глухов, Д.О. Локально-эволюционный метод составления учебного расписания вуза / Д.О. Глухов, А.О. Глухов, О.Н. Травкин // Вестн. Полоц. гос. ун-та. Серия С. Фундаментальные науки. – 2005. – № 10. – С. 168–178.
7. Деканова, М.В. Математическая модель и алгоритм построения расписания учебных занятий университета / М.В. Деканова // Вестн. Полоц. гос. ун-та. Серия С. Фундаментальные науки. – 2013. – № 12. – С. 24–33.

Поступила 05.03.2014

METHODS OF SOLVING MULTICRITERIA PROBLEM SCHEDULE OF TRAINING SESSIONS UNIVERSITY

M. DEKANOVA

The existing approaches to solving the problem of scheduling such as locally-evolutionary method, multi-agent approach, intellectual method, a method of successive analysis of variants, genetic algorithm, a method of coloring of graphs have been analyzed. The advantages and disadvantages of them have been emphasized. As output the problem of scheduling refers to the class of the activation discrete problems with a finite set of alternatives. The solution of the scheduling problem is complicated with the multicriteriality and the multivariance. It has been shown that it is necessary to apply the integrated approach reckoning in the advantages of all the considered in this work methods.