

## ВОПРОСЫ АЛГОРИТМИЗАЦИИ СОСТАВЛЕНИЯ ВРЕМЕННОЙ СЕТКИ РАСПИСАНИЯ УЧЕБНЫХ ЗАНЯТИЙ

И. Э. НААЦ

(Представлена научным семинаром вычислительной лаборатории)

В общей задаче алгоритмизации процесса составления расписания учебных занятий можно выделить ряд самостоятельных вопросов. Это прежде всего такие задачи, как распределение учебной нагрузки по преподавателям, составление временной сетки расписания и распределение аудиторного фонда. Первая задача находится несколько в стороне от проблемы автоматизации составления расписания и в данной работе не рассматривается. Решение ее связано с вопросами научной организации и планирования педагогической работы преподавателей как одной из специфических форм деятельности человека. К сожалению, эта область исследований мало привлекает внимание специалистов, и в настоящее время отсутствуют какие-либо рекомендации на этот счет.

Вторая задача является самой значительной по затратам ручного труда составителя. Исследованию вопросов составления временной сетки расписания и разработке алгоритмов для автоматизированного составления на вычислительных машинах и посвящена данная статья. Третья задача является предметом самостоятельного исследования и в настоящей работе не рассматривается.

### Основные понятия. Определение логической структуры алгоритма составления временной сетки расписания

Прежде чем приступить к анализу и определению структуры алгоритма, необходимо ввести понятия, в терминах которых возможно формальное и однозначное описание всей последовательности операций по составлению временной сетки расписания учебных занятий. Введем понятие элемента расписания.

Определение 1. Элементом расписания будем называть выражение вида

$$A\{abc\dots\}.$$

В общем случае символы  $A$ ,  $a$ ,  $b$ ,  $c\dots$  могут иметь буквенное, буквенно-цифровое и цифровое обозначение.

$A$  называется индексом элемента, выражение в фигурных скобках — характеристикой элемента. Индекс выражает код преподавателя, а ха-

рактеристика — группы, в которых он ведет данное занятие. Например, запись

$$A\{12345678\}$$

означает, что преподаватель  $A$  проводит занятие одновременно в восьми группах, номера которых условно обозначены цифрами 1, 2, ... 8. Запись  $B\{00045000\}$  означает, что преподаватель  $B$  проводит одновременно занятие с группами 4 и 5 из восьми рассматриваемых.

Если все символы характеристики, за исключением одного, есть 0, то это значит, что преподаватель с данным индексом проводит занятие только в одной группе, номер которой и отличен от нуля. Такие элементы, как правило, обозначают практические занятия. Элементы, обозначающие точные занятия, имеют по крайней мере два ненулевых символа. Таким образом, для построения характеристики элемента используется позиционный принцип, т. е. отсутствие занятия или его наличие в данной группе обозначается расположением нулевого символа или номера группы в определенном месте характеристики. В связи с этим можно дать другое чисто семантическое определение элемента.

Определение 2. Под элементом расписания понимается символическая запись в цифро-буквенной форме того факта, что преподаватель  $A$  имеет, по крайней мере, одно занятие с любой из  $N$  рассматриваемых групп.

Определение (1) дает нам структуру элемента, определение (2) — его содержание.

В результате вся исходная информация об учебной нагрузке в  $N$  группах может быть представлена в виде упорядоченного множества (колонки, таблицы и т. д.) элементов. Пример такой записи

$$\begin{array}{l} A \{12300000\} \\ B \{10345000\} \\ C \{00000678\}. \\ \dots \end{array}$$

Если преподаватель проводит одно и то же занятие в данной группе несколько раз за период времени, для которого составляется расписание (обычно этот период охватывает две недели), то в колонку нагрузки соответствующий элемент записывается соответствующее число раз.

Определение 3. Два элемента будем называть совместными, если они имеют разные индексы и разные характеристики, в противном случае элементы будем называть несовместными.

Условие совместности, данное в определении (3), имеет исключительно важное значение для дальнейшего анализа, поэтому сделаем более подробные пояснения.

Существо процесса составления временного расписания учебных занятий сводится фактически к выявлению таких занятий, которые могут проходить одновременно или, иными словами, могут быть совмещены во времени. Очевидно, что к таким занятиям относятся занятия разных преподавателей в разных группах. В противном случае имеет место так называемое наложение во времени, и расписание теряет практический смысл. В введенных выше терминах составление расписания может быть представлено как разбиение исходного множества элементов на подмножества совместных элементов. Такие подмножества в дальнейшем будем называть блоками, т. е. имеет место следующее:

Определение 4. Блок — это множество, все элементы которого совместны между собой.

Отметим, что введенное понятие совместности отлично от понятия эквивалентности. Поэтому задача разбиения исходного множества элементов на блоки отлична от разбиения множества на классы эквивалентности. Действительно, справедливы следующие леммы:

Лемма 1. Элемент  $A$  несовместен самому себе.

Лемма 2. Если элемент  $A$  совместен элементу  $B$ , а элемент  $B$  совместен элементу  $C$ , то  $A$  и  $C$  необязательно совместны между собой.

Эти леммы легко доказываются от противного. В понятии же эквивалентности элемент множества эквивалентен самому себе. Это говорит о том, что условие совместности имеет более сложную логическую структуру. Из вышеприведенного анализа следует, что для того, чтобы решить, можно включить рассматриваемый элемент в данный блок или нет, необходимо проверить выполнение условия совместности со всеми элементами этого блока, т. е. в целом по блоку. При разбиении же множества на классы эквивалентности достаточно проверить условие эквивалентности рассматриваемого элемента с любым элементом данного подмножества.

В блоке как множестве сложной структуры можно построить два простых множества: множество всех индексов элементов данного блока и множество всех характеристик. Эти множества не содержат двух одинаковых элементов, если не считать элемент 0. При построении второго множества (объединение характеристик) сохраняем позиционный принцип расположения элементов, отбрасывая соответствующие нулевые элементы и сохраняя тем самым неизменным общее число символов (разрядов) при переходе от характеристик к данному множеству. Эти построения используем для упрощения логической структуры алгоритма формирования блоков из совместных элементов расписания. С этой целью введем понятие гиперэлемента блока.

Определение 5. Гиперэлемент блока есть выражение вида

$$\gamma ABC... \{abc...\},$$

где  $\gamma$  — номер гиперэлемента (соответственно блока),

$ABC \dots$  — индекс гиперэлемента,

$\{abc \dots\}$  — характеристика.

Характеристику гиперэлемента, как и характеристику элемента, ниже будем записывать в цифровой форме. При этом индекс гиперэлемента и его характеристика являются объединением соответственно индексов и характеристик по вышеуказанному правилу всех элементов, входящих в блок с номером  $\gamma$ .

Лемма 3. Гиперэлемент не является элементом блока.

По определению гиперэлемент является объединением индексов и характеристик элементов и, следовательно, является несовместным к любому элементу данного блока. Это и доказывает лемму (3).

Лемма 4. Если блок состоит из одного элемента, то этот элемент является гиперэлементом этого блока. Справедливость леммы (4) вытекает из определения (5) гиперэлемента.

Гиперэлемент является некоторой обобщенной характеристикой блока, не имеющего содержательного смысла и предназначенного для упрощения логической структуры алгоритма. Более полное назначение гиперэлемента в алгоритме машинного составления расписания будет раскрыто ниже, а пока сформулируем следующую теорему.

Теорема 1. Если элемент совместен некоторому гиперэлементу, то он совместен соответствующему блоку в целом.

Теорема легко доказывается от противного с использованием леммы (3).

Полученные результаты можно сформулировать следующим образом: для того, чтобы решить, можно включить данный элемент в блок  $\gamma$  или нет, достаточно проверить условие совместности данного элемента с гиперэлементом этого блока. Таким образом, введение гиперэлемента значительно сокращает число операций, выполняемых машиной, поскольку отпадает необходимость проверки условия совместности данного элемента с каждым элементом блока.

### Полные и неполные блоки. Заполнение временной сетки расписания

Определение 6. Характеристику, не содержащую нулей, будем называть полной относительно множества групп, в противном случае — неполной.

Примером полной характеристики элемента (соответственно гиперэлемента) является характеристика вида  $\{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\}$  и неполной  $\{1\ 0\ 0\ 0\ 0\ 0\ 0\ 8\}$ .

Определение 7. Если гиперэлемент имеет полную характеристику, то соответствующий блок будем называть полным относительно рассматриваемого множества групп, в противном случае — неполным.

Полнота блока относительно рассматриваемого множества групп означает, что в данный блок уже не может быть включен ни один элемент из исходного множества занятий и процесс формирования данного блока должен быть окончен. Справедлива следующая

теорема 2. Если характеристика какого-либо элемента исходного множества занятий полна, то этот элемент образует полный блок.

Иными словами, теорема (2) определяет условия, при которых полный блок состоит из одного элемента. Этот единственный элемент является в то же время и гиперэлементом блока.

Предположим, что составляется расписание для  $N$  групп, при этом  $i$ -тая группа за две недели имеет  $a_i$  занятий (соответственно  $2 a_i$  часов) ( $i = 1, 2, \dots, N$ ). Если обозначить через

$$a_{\min} = \min_{1 < i < N} a_i, \quad (1)$$

$$a_{\max} = \max_{1 < i < N} a_i, \quad (2)$$

то число полных блоков равно

$$n_{\text{п}} = a_{\min} \quad (3)$$

и соответственно неполных

$$n_{\text{н}} = a_{\max} - a_{\min}. \quad (4)$$

Общее число блоков определяется как

$$n = n_{\text{п}} + n_{\text{н}}. \quad (5)$$

Величина  $n$  определяет число ячеек временной сетки, необходимое для проведения занятий в этих  $N$  группах. При этом под временной ячейкой в данном случае понимается отрезок времени, который необходим для проведения одного занятия, т. е. 2 часа. Тогда общее рабочее время за 2 недели (ежедневно по 7 занятий с 9 часов утра до 11 часов вечера)

в  $2 \times 7 \times 12 = 168$  часов может рассматриваться как временная сетка из 84 ячеек. В этом смысле далее и будут использоваться понятия временной сетки и ячейки.

В этих понятиях дальнейшее составление расписания после формирования блоков состоит в размещении их по ячейкам временной сетки. При этом, если рассматривается  $N$  групп, то в одну и ту же ячейку не должны попасть два блока. Условием непересечения двух блоков в одну и ту же ячейку является их несовместность.

Определение 8. Два блока будем называть несовместными, если несовместны их гиперэлементы. В противном случае — совместными.

Лемма 5. Для  $N$  рассматриваемых групп все блоки несовместны друг с другом.

Действительно, если бы нашлись два совместных блока, то они могли бы быть объединены в один ввиду совместности их гиперэлементов. В частности, это доказывает справедливость (5), так как  $a_i$  определяет, сколько раз  $i$ -тая группа входит в элементы исходного множества занятий.

В результате распределения блоков по временной сетке каждому блоку присваивается номер соответствующей ячейки, и временное расписание можно считать составленным. Однако практически распределение блоков по ячейкам весьма сложно и трудоемко.

Рассмотрим два множества групп  $G$  и  $P$ . Множество преподавателей, ведущих занятия в этих группах, обозначено через  $I_G$  и  $I_P$  соответственно. Справедлива следующая

теорема 3. Блоки множества  $G$  совместны блокам множества  $P$  только в том случае, когда множества  $I_G$  и  $I_P$  не пересекаются, т. е.

$$I_G \cap I_P = \emptyset.$$

Теорема доказывается от противного с использованием соответствующих определений 1.

В результате возникает следующая интересная задача: можно ли все множество групп разбить на систему подмножеств так, чтобы соответствующие подмножества преподавателей не пересекались. Если такая задача решается, то для каждого из полученных подмножеств групп временное расписание можно составлять независимо друг от друга, ибо блоки их будут совместны. Это очень важно с той точки зрения, что при наличии таких независимых множеств групп количество одновременно перерабатываемой информации будет сравнительно небольшим. В противном случае приходится оперировать большим объемом информации, даже если расписание составляется для небольшого числа групп. Если же не удастся разбить группы на независимые подмножества, то ставится задача совмещения блоков для множеств  $G$  и  $P$ . Поскольку множества блоков  $L_G$  и  $L_P$  содержат несовместные блоки, то задача сводится к нахождению пары совместных блоков: одного из  $L_G$ , другого из  $L_P$ . Для решения этой задачи может использоваться тот же алгоритм, что и для составления блоков.

### Компактные расписания

Выше уже отмечалось, что число временных ячеек, необходимое для размещения расписания, определяется числом составленных блоков, поскольку в каждую ячейку помещается по одному блоку. В свою очередь число блоков определяется максимальным значением нагрузки  $i$ -той группы из множества всех групп. Однако анализ будет неполным, если

не учитывать множество преподавателей при формировании блоков.

Пусть, как и прежде,  $a$  — число занятий в  $i$ -той группе за две недели, где  $i=1, 2 \dots N$ . Введем  $b_j$  как число занятий, проводимых  $j$ -тым преподавателем в  $N$  рассматриваемых группах тоже за две недели, где  $j=1, 2 \dots M$ .

Вполне очевидно выполнение следующего неравенства

$$\sum_{j=1}^M b_j \leq \sum_{i=1}^N a_i. \quad (6)$$

Равенство (6) выполняется только для множества практических занятий, т. е. когда каждый преподаватель одновременно ведет занятие только с одной из групп. В общем случае с учетом лекции (6) дает строго неравенство. Поскольку в каждый блок преподаватель и группа входят только по одному разу, то справедлива следующая

теорема 4. Число блоков определяется как

$$n = \max_{1 < i < N} a_i, \quad (7)$$

если

$$\max_{1 < j < M} b_j \leq \max_{1 < i < N} a_i, \quad (8)$$

и как

$$n' = \max_{1 < j < M} b_j, \quad (9)$$

в противном случае, т. е. когда

$$\max_{1 < j < M} b_j > \max_{1 < i < N} a_i. \quad (10)$$

В первом случае, т. е. когда выполняется условие (8), расписание будем называть компактным, во втором — некомпактным.

Компактность означает, что при заполнении блоками временных ячеек соответствующее расписание не будет содержать «окон» для групп. Кстати, это является одним из основных требований, предъявляемых к расписанию учебных занятий.

При некомпактном расписании, поскольку число блоков  $n'$  превышает все без исключения числа  $a_i$ , при заполнении временных ячеек блоками в расписании возможны «окна». Этот факт объясняется либо нехваткой преподавателей, либо неравномерным распределением нагрузки, так как наибольшая нагрузка, приходящаяся на какого-то преподавателя (или преподавателей), превышает наибольшую учебную нагрузку группы согласно (10). Такие неравномерности в распределении нагрузки должны быть устранены по крайней мере так, чтобы

$$\max_{1 < j < M} b_j = \max_{1 < i < N} a_i, \quad (11)$$

ибо в противном случае произойдет неоправданное удлинение расписания на  $\Delta n = b_{\max} - a_{\max}$  ячеек, или  $2\Delta n$  часов. Однако подобное удлинение расписания вряд ли возможно в практических случаях. В качестве примера анализа рассматривалось исходное множество занятий для

групп геологоразведочного факультета по графику 1965/66 учебного года.

Анализ показывает, что каждые 8 групп (среднее число групп на курсе) в среднем при 15 различных предметах «обслуживают» 46 и более преподавателей, т. е. имеет место слабая локализация преподавателей в данных группах. Это заставляет сомневаться в возможности разбиения всего множества групп на независимые (в смысле теоремы 3) подмножества с целью уменьшения объема одновременно перерабатываемой информации. Поэтому, чтобы получить соответствующую эффективность в применении вычислительных средств для составления временной сетки расписания, необходимо составлять его сразу в целом для всего института. В противном случае промежуточная ручная обработка информации по увязке отдельных частей сетки поставит под сомнение целесообразность использования вычислительных машин.

### Алгоритм составления временной сетки расписания

Вначале дадим общее описание алгоритма. В целях простоты будем считать, что элемент расписания помещается в одну ячейку памяти машины, при этом принимается следующая структура ячейки:

$$\theta a, \gamma a, \tau a,$$

где в часть ячейки  $\tau a$  записывается индекс преподавателя, в  $\gamma a$  — характеристика элемента и в  $\theta a$  — признаки «элемент занят» или «элемент свободен», т. е. вошел или нет в один из сформированных ранее блоков. Исходное множество занятий предстанет в виде множества ячеек  $A[a+i]$  ( $i=1, 2 \dots N$ ). Индекс гиперэлемента будем формулировать в ячейке  $r1$ , а его характеристику — в ячейке  $c1$ . Если элемент совместен гиперэлементу блока, формируемого в данный момент времени, то его характеристика  $\gamma[a+i]$  складывается с содержимым ячейки  $c1$ , а индекс заносится в соответствующее место ячейки  $r1$ . При этом осуществляется печать (выход) содержимого ячейки  $(a+i)$ . В результате блок печатается в виде колонки совместных элементов. Блок считается сформированным, если полна характеристика гиперэлемента, т. е.  $c1 = 111\dots$ , либо среди оставшихся исходных элементов больше нет совместных.

Ячейки временной сетки представлены массивом ячеек памяти НМЛ  $B[b+j]$ . В ячейке  $(b+j)$  записаны индексы преподавателей порядком  $\tau$ , которые уже ранее при составлении расписания были помещены в данную временную ячейку. Если сформированный блок может быть помещен в данную временную ячейку (соответственно ячейка памяти  $(b+j)$ ), т. е. если содержимое ячейки  $(b+j)$  и  $r1$  совместно, то вслед за отпечатанными элементами блока печатается адрес ячейки  $(b+j)$ , соответствующий номеру временной ячейки. Индексная часть гиперэлемента  $r1$  заносится в ячейку  $(b+j)$ . Затем программа повторяется для оставшихся элементов исходного множества занятий.

Формальное описание алгоритма дадим в терминах алгоритмического языка АЛГОЛ-60.

### Алгоритм

Начало массив  $A[a+i]$ ;

L:  $i:=1$  перейти к Q;

Q: если  $\theta[a+i]=0$  то выполнить;

Начало, если  $c1=0$  то

$c1:=\gamma[a+i];$

Вывод  $(a+i);$

$\Theta[a+i]:=1;$

$\Gamma1:=\tau[a+i];$

$\kappa:=1;$

Иначе перейти к  $M;$

Конец

Иначе выполнить

Начало  $i:=i+1;$

Если  $i \leq N$ , то перейти к  $Q$  иначе;

(останов)

Конец

$M$ : начало если  $\gamma[a+i] \neq c1$  то перейти к  $P$

Иначе выполнить;

$i:=i+1;$

Если  $i \leq N$  то  $Q$  иначе  $L1;$

Конец

$P$ : индекс:  $=\tau[a+i] \times 2^{\uparrow}(\tau \times j);$

Если индекс  $\neq r1$  для  $j=0$  шаг 1 пока  $k$ , то

$c1:=c1+\gamma[a+i];$

$r1:=r1+\tau[a+i] \times 2^{\uparrow}(i \times k);$

Вывод  $(a+i);$

$\Theta[a+i]:=1;$

$k:=k+1;$

Перейти к  $P1$ , иначе  $L1;$

$P1$ : если  $c1=11\dots$ , то  $L1$  иначе

$i:=i+1$  перейти к  $Q;$

$L1$ : массив  $B[b+j]$

$j:=1;$

$Q1$ : если  $\Theta[b+j]=0$  то  $G1$  иначе

$j:=j+1$  перейти к  $Q1;$

Комментарий: подразумевается, что  $j \leq M$  практически для всех случаев, условие  $j > M$  означало бы, что данный блок не может быть помещен ни в одну временную ячейку. Если же необходимо осуществлять контроль заполнения временной сетки, то можно ввести в  $Q1$  оператор:

если  $j \leq M$  то  $G1$  иначе;

Затем следует останов, ибо возникает неопределенность в размещении блока, которая может быть устранена увеличением числа ячеек временной сетки.

$G1$ :  $p:=0$

$G2$ : начало лектор:  $=\tau[r1] \times 2^{\uparrow}(\tau \times p);$

время:  $=\tau[b+j] \times 2^{\uparrow}(\tau \times q);$

Для  $q=0$  шаг 1 пока  $t$  выполнить;

Если лектор  $\neq$  время то

$p:=p+1;$

Если  $p \leq k$  то G2 иначе G3 иначе;

$j := j + 1$  перейти к Q1;

Конец;

G3:  $\Theta[b+j] := 1$ ;

выход  $(b+j)$

Комментарий, т. е. печатается время для данного блока.

$c1 := 0$ ;

$r1 := 0$ ;

$k := 0$ ;

Перейти к L

Конец

Описанный алгоритм был реализован на машине «Минск-1» в вычислительной лаборатории ТПИ (см. [1]).

#### ЛИТЕРАТУРА

1. Н. М. Былино. Разработка программы составления временной сетки расписания на ЭВМ «Минск-1». (Настоящий сборник).

---