The application of a digital computer to the construction of timetables

By Elizabeth D. Barraclough*

A method of constructing school timetables by computer is described. Hand methods are adapted where appropriate and supplemented when necessary. A theoretical basis for the different methods of allowing a choice of sixth-form subjects is also discussed.

In Britain the main concern for the construction of timetables is with the comparatively small-scale but complex problem arising in schools. British Universities are generally smaller than American ones, and also tend to arrange their teaching within departments so that the problem of many thousands of students all choosing from the same courses does not often arise. However, it is worth considering the methods used for the large-scale problem since it is in this field that most of the computer work has been done. Both the problems of timetable construction and student scheduling have been tackled by Holz (1963) at the Massachusetts Institute of Technology. The method used assumes the course structure found in most American Universities. In each subject several independent courses are given, so that the total number of courses offered is large. Ideally it should be possible to take any course with any other course, but in practice this can rarely be achieved. In order to make the choice as wide as possible each course can be repeated at a different time. It may also be necessary to cater for large numbers of students and to split the course into several classes taking place at the same time. The distribution of periods is done by allocating time-patterns to a course, for example one requiring three periods could be allocated the time-pattern Monday, Wednesday and Friday, 9.00 a.m.–10.00 a.m. The week, then, is split into definite time-patterns not all of which are independent. Thus two courses with different time-patterns could still clash with each other.

The requirements for each course are given as the number of classes, together with possible instructors and time-patterns listed in order of preference. At each step in the compilation the best available time-patterns and instructors are allocated. If there are not sufficient instructors available without causing a clash the requirement remains unsatisfied and is marked appropriately. Thus the final timetables may not contain all the required classes. In a large college this does not make the timetable useless since it is prepared on the basis of probable, rather than actual, student selections. The separate process of scheduling students to classes in the compiled timetable is also described in the *GASP* manual (Holz, 1963). The method used is quite simple. Each student selects several courses and these selections are entered into the program. Initially the selection for a student is checked to see if it is compatible with the timetable; if not, the student cannot be scheduled and a message to this effect is printed. The next stage is to compute a workable schedule; the "value" of the schedule is computed to determine whether it is a good one or not. Output occurs if the schedule is acceptable; if not a new schedule is computed and tested. A limit is imposed on the number of schedules tried, and when this limit is reached the best schedule computed is printed. Each schedule is considered only on its own merits without reference to any other student's schedule. Thus if a student is scheduled badly and fills the last available place in a class this step cannot be retraced.

A slightly different approach to this problem has been made by Sherman (1958) at Purdue University, where the number of students is about twenty thousand. With this number the time to schedule each student is of major importance. The method used aims to schedule each student as quickly as possible by reducing the numbers of schedules tried before a satisfactory one is found. This can be achieved by ordering the classes so that the most difficult class to fit occurs first in the student selection. Several factors can be brought in to this ordering process. The most obvious one is the number of classes available to the course. Clearly if only one class is available then the student can either be fitted into this class or not, so that rejection occurs very early. Another factor taken into account by this program is the distribution of students within the classes. By keeping the numbers fairly even fewer classes reach their full capacity, and thus late registrants can be fitted into the timetable more easily.

The use of a computer for the large-scale problem seems well justified. The question is whether the same rewards can be found in the realm of the small school timetable.

The simple timetable problem

In its simplest form the problem is to construct a timetable for each master and class that satisfies the teaching requirements and obeys the essential restriction

* *University Computing Laboratory, Newcastle upon Tyne* 2.

that no master or class is required in two places at once. The teaching requirements can be stated as the number of periods during the week that each master shall meet each class. As an example consider the problem of a "school" with four masters, four classes and four periods per week, each master being required to teach each class for one period in the week. The masters and classes are thus fully occupied. For this example a Latin square solution can easily be found; one of the many solutions is shown in **Fig. 1.**

The same simple problem can be used to show how difficulties can arise. Suppose that the timetable was being compiled and the partial timetable shown in **Fig. 2** was formed. Clearly the remaining requirements cannot be inserted without retracting at least one step.

The practical timetable

In the timetable for a real school the requirements are not normally restricted to one master teaching one class. One type of complexity that can arise is caused by all the classes in one year having games together, leading to the requirement that one master shall teach several classes. In this situation it is easy to construct an impossible set of requirements. In the simple example using four masters and four classes, if classes 1 and 2 had to be taught for two periods by master 3, classes 2 and 3 for two periods by master 4, and classes 1 and 3 for two periods by master 3 then clearly a solution is impossible.

Any proposed method must consider the three queries raised by these examples.

(1) Is a solution possible?
(2) Is the partial timetable compatible with the remaining requirements?
(3) Which of the possible solutions is most acceptable?

Solution by calculation

A method, proposed by Appleby, Blake and Newman (1961), attempts to calculate the best position for each entry before inserting it in the timetable. The method uses two parameters for each requirement.

(1) The difference between the number of possible positions for this entry and the number of required positions.
(2) The difference between the number of masters and classes required for those requirements where parameter (1) is zero.

If the first parameter is negative no solution is possible. For positive values the requirement with the lowest value of this parameter is chosen for insertion.

In order to avoid the situation where the remaining requirements are incompatible with the partial timetable, the calculation of parameters is extended for some selected groups of requirements.

Although this method can answer all the queries it does have a severe practical limitation due to the time taken to perform the extensive calculations before any entry can be inserted in the timetable. The final program only made 3 or 4 entries per minute on the ACE computer at the N.P.L., which for a school of thirty masters and thirty-five periods would take approximately two hours to complete the timetable. A further drawback is that despite these calculations it is still possible to reach a failure situation from which the only recovery is to return to an arbitrary point in the compilation and restart.

CLASS \ PERIODS	1	2	3	4
1	1	2	3	4
2	4	3	2	1
3	2	1	4	3
4	3	4	1	2

Fig. 1.—Class timetable showing the masters teaching at each period

CLASS \ PERIODS	1	2	3	4
1	1	2	3	4
2	2	3	4	1
3		4	1	3
4	3	1	2	

Fig. 2.—Partial timetable that cannot be completed

A theoretical method

Another possible method (Gotlieb, 1962; Csima and Gotlieb, 1964) is a theoretical extension of the previous method for a limited class of problem. The problem considered was restricted to requirements involving only one master and one class, and was reduced in size by compiling the timetable for each day rather than for the whole week at once. In this way it becomes feasible to test all possibilities of master and class groups instead of a few specific groups as in the previous method.

The main advantage of this method is that it completely answers both the first two queries and, by splitting the weekly requirements into daily ones, evades the third. However, it only seems to succeed by ignoring the complex requirements which appear to be the major difficulty in most timetable compilations. A further drawback with this method, as with the previous one, is the amount of time taken to produce a timetable. A very simple problem with nine masters, nine classes and nine periods where masters and classes were fully occupied took 1·8 minutes on an IBM 7090 computer. The same problem was tried on the much smaller and slower Pegasus computer with the program to be described and a solution produced without taking any appreciable calculation time.

Compilation by hand

The hand method of compilation is described by Lewis (1963). Its attraction is its proved ability to produce a timetable. The fundamental difference from the previous approaches is the possibility of recovery from an incompatible partial timetable by interchanging entries. The method, however, gives no guarantee of finding a solution.

The choice of method for compiling a school timetable seems to lie between the theoretical method and the hand method. Although the theoretical method has the advantage of finding a solution if one exists, it also has the prohibitive restriction that the complex requirements found in a normal school cannot be tackled.

The method to be described is therefore based primarily on the hand method. A similar approach has been programmed for a Bull Gamma 30 computer by Berghius, van der Heiden and Bakker (1964). They have made certain simplifications to the method as used in this country, but nevertheless have shown that it is a practical possibility to simulate a hand compiler on a computer. Our work, which evidently proceeded concurrently with that in Holland, incorporates many more of the complications arising in school timetables in this country. The hand technique is followed fairly closely, though naturally additions have been made to exploit the facilities of a computer and thus improve the compilation method.

A school and its timetable

In order to construct the timetable for a particular school, it is necessary to know both the characteristics of the school and the requirements for and restrictions on masters and classes. A complete specification for a school needs to contain the following.

(1) The number of masters.
(2) The number of classes.
(3) The number of periods each day.
(4) The number of days each week.
(5) The breaks during the day that double periods must not span.
(6) Special features such as triple periods, T.V. lessons or games.

Master and class requirements

These are the requirements that specify the subject content of the new timetable. They must present to the program all the details of the curriculum. For most schools the requirements consist of a selection from the following categories.

(1) A single master teaches a single class.
(2) A choice of subjects exists within a class.
(3) All the forms in the same year take a particular subject together.
(4) A selection of options is made in the sixth form.
(5) A complete choice of subjects by individual pupils is made in the sixth form.

A single master teaching a single class is the simplest of these requirements. For example, if master 11 was required to teach seven periods of mathematics to class 6 where two of the seven periods were to be a double period, this could be stated thus: Master 11 shall teach mathematics to class 6 for 5 single periods and 2 periods in a double period. The fact that the subject to be taught is mathematics does not affect the construction of the timetable. Very often the subject is determined uniquely by the master. Where this is not the case, the subject to be taught is easily inserted after the timetable is completed. The requirement can therefore be abbreviated to a form suitable for input to the program:

11, 6, 5, 2

Where a choice of subjects is possible for a class, this implies that two or more masters must be available at the same time to teach the appropriate subjects. A typical example in a mixed school occurs when the boys take woodwork and the girls needlework. If master 7 teaches woodwork and "master" 3 needlework to class 2 for 4 periods a week in two double periods, this requirement can be stated thus: Master 3 and master 7 shall teach class 2 for 4 periods in double periods each week. In abbreviated form this could read:

(3, 7) 2, 0, 4

The zero indicates that there are no single periods. The brackets round the masters indicate that both masters must be available at the same time. Such a group of masters is called a *set*.

The idea of a set can be extended to cover the case where the pupils in one year are split into different classes for a particular subject such as French. In this situation the whole "year" must be taught French at the same time. If the year, consisting of classes 4, 5 and 6, requires five single periods of French each week and masters 5, 12 and 17 are allocated to teach them, the requirement could be stated thus:

(5, 12, 17) (4, 5, 6) 5, 0

Here there are two groups, one of masters 5, 12 and 17 which will occur as a set in the class timetables, and another of classes 4, 5 and 6 which will occur as a set in the master timetables.

In order to allow television or radio broadcasts to be heard in a particular lesson, it must be possible to fix entries in the timetable. For example, if a French broadcast that is suitable for form 3 takes place at 11.00 a.m. on Wednesday, then form 3 must have a French lesson at that time as a first priority.

With the present form of master and class requirements there is no means of specifying the time at which any lesson should take place. Accordingly some other method must be used to insert fixed entries. The system adopted is to provide an initial timetable for each master which contains any fixed entries. The remainder of this timetable can either be free periods or be part of last year's timetable. It is essential that the timetable should be consistent, that is, no master or class is expected in two places at once.

The method of construction

In the method of compiling timetables by hand, fixed entries such as T.V. lessons are inserted first, then the entries with the most complicated requirements such as sets and double periods. Finally single periods are entered until an impossible entry is reached. At this stage an attempt is made to interchange entries so that the required master and class are free at the same time. The computer approach adopted is similar.

The main difference between the program and hand methods is that the former attempts to keep part of the previous year's timetable as a basis for the new one. The labour involved makes it nearly impossible to do by hand. When a school timetable is first attempted on a computer no timetable exists on paper tape. Although this can be created from last year's hand-compiled timetable it is probably not worth the effort involved. Once a timetable has been produced by the computer, however, it is worth while trying to retain some of it in the new timetable.

The program

An outline of the program is evident from the flow diagram shown in **Fig. 3.** During the construction of the timetable its current state is recorded by

(1) the master timetable so far created,
(2) a binary digit pattern showing the periods available for each master,
(3) a binary digit pattern showing the periods available for each class,
(4) definitions showing the classes implied by each set number in the master timetable.

Making entries in the timetable

Single periods: one master teaching one class for single periods only.

In the list of requirements several of the form m_i, c_j, s, 0 (demanding master i teach class j for s single periods) can appear. In order to assign these single periods it is necessary to know those periods when master i and class j are both free, and then choose s of them.

The common periods available are described in the binary pattern formed by logical 'and' of the available digits for master i and the available digits for class j. In abbreviated notation this "and" operation could be written $pm_i \wedge pc_j$; for example, if the available digits for master i and class j for one day were

master i	1101001
class j	1000111

the logical "and" would give 1000001 showing that both

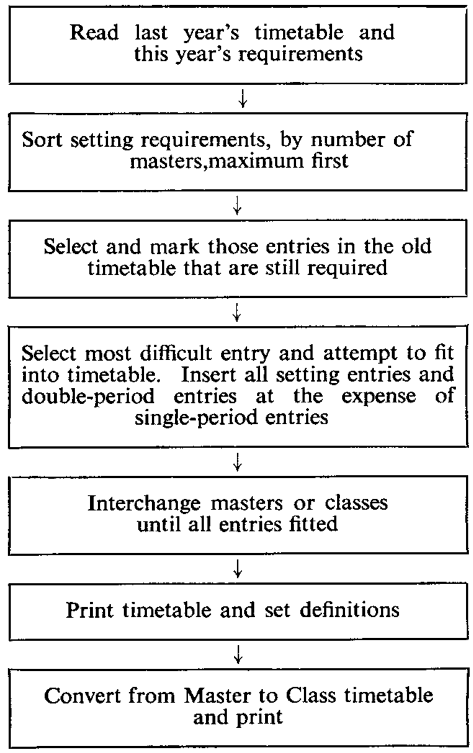

Fig. 3.—Timetable program: schematic diagram

master and class were available in the first and last periods but no other.

From the common available digits s are chosen by the simple rule of taking the first available one each day, returning to the beginning of the week if necessary. Clearly more complicated rules could be inserted here if desired. A set of period-required digits is then formed. For each of these required digits an entry is made in the master's timetable showing that class j is to be taught. The required digits are also removed from the available digits for both the master and class.

Double periods: one master teaching one class for double periods only.

To make an entry corresponding to the requirement m_i, c_j, 0, d, the same "and" $pm_i \wedge pc_j$ is performed. Pairs of digits that do not cross a break must be found by referring to the pattern showing when breaks occur. This pattern of digits, denoted by de, shows where double periods may end. The digits showing where double periods may start can be found by the following logical operations:

$pm_i \wedge pc_j \wedge de$ which give a "1" where the second period is free. Let $(pm_i \wedge pc_j \wedge de)'$ denote the same pattern of binary digits but shifted up one digit position. The 1s now occur at positions which could be the start of a double period. Then $(pm_i \wedge pc_j \wedge de)' \wedge (pm_i \wedge pc_j)$ gives a 1 where a double period may start.

For example, if the binary digit patterns for one day were

$$\begin{aligned} pm_i &= 1110111 \\ pc_j &= 0110110 \\ de &= 0101011 \\ pm_i \wedge pc_j &= 0110110 \\ \text{then} \quad (pm_i \wedge pc_j \wedge de) &= 0100010 \\ (pm_i \wedge pc_j \wedge de)' &= 1000100 \\ (pm_i \wedge pc_j \wedge de)' \wedge (pm_i \wedge pc_j) &= 0000100 \end{aligned}$$

showing that a double period may start in the fifth period, but rejecting the two consecutive periods in the second and third periods since they cross a break.

From these digits showing where double periods may start $d/2$ are chosen in the same way as the required digits for single periods. For each of these required digits an entry is made in the timetable for master i showing that he is teaching class j for a double period. The entry is also inserted for the second period of the double period, so that any entry in the timetable contains within itself sufficient information for the program to identify it. Both digits corresponding to each double period are removed from the available digits for the master and class.

Setting periods

Setting periods can be tackled in exactly the same way once the common available digits have been found. This can be done very simply by extending the "and" operation to include all the masters and classes required. For example, if the requirement was (m_i, m_h, m_g) (c_j, c_k) s, d, then the common available digits would be given by $pm_i \wedge pm_h \wedge pm_g \wedge pc_j \wedge pc_k$.

From these available digits the selection of the required digits is done as for the single periods. The entry made in the timetables for all the masters in the set shows the code number of the set of classes being taught. The required digits are, of course, removed from the available digits of all the masters and classes.

During the second stage of the program the timetable is built up entry by entry using these basic operations. The most difficult entries such as sets and double periods are given preference over single periods. This preference is executed in two ways. Firstly the sets are attempted before the single or double periods, and within the set requirements a selection procedure is used to determine the most awkward set to fit; this is then inserted first. Secondly an attempt is made to remove single periods remaining from last year's timetable if these obstruct a set or a double period.

In order to fit an entry where no common period-available digits exist for all the masters and classes, some "available" digits have to be found which satisfy as many of the masters and classes as possible. From these "available" digits the required digits are selected in the usual way, but for any masters or classes not free the corresponding entry in the timetable is tested. If the entry is a single period then the required digit is accepted and the single period overwritten, but if the entry is a set or a double period then the required digit has to be rejected. The whole process may be repeated several times, perhaps only fitting one period each time, until all the requirements are satisfied or a complete failure occurs. A method of interchanging set entries and thus avoiding failures at this stage is discussed later.

Insertion of the final entries

The third and final section of the program is concerned with fitting all the entries that have been rejected or could not be inserted during the second stage. At this point all the remaining requirements are single entries, since a failure would have occurred if this was not the case. Attempts are made to insert these entries by any necessary interchanges of masters or classes in the timetable. Throughout these attempts only one period is considered at a time, so that the requirements are treated as if they were of the form m_i, c_j, 1, 0. Where more than one period is required the process is merely repeated.

Several methods of inserting the entries are possible and they are attempted in increasing order of complexity.

(*a*) First the straightforward method is used since some of the requirements displaced by sets may have some common p.a.d. (period available digit).

(1) Find any common p.a.d. $pm_i \wedge pc_j$.
(2) Select the first period in the week.
(3) Insert the entry in the timetable and reduce the p.a.d. for m_i and c_j.

(*b*) When there are no common p.a.d. the next possibility is to make an interchange within the timetable for m_i. This interchange should be made so that m_i is freed for one of c_j p.a.d. If such an interchange can be found then the entry can be inserted and no other entry is rejected. To make the correct interchange a class entered in the timetable for m_i must have a free period corresponding to one of m_i p.a.d. One of the entries for this class must also occur at one of c_j p.a.d. The steps are:

(1) Form a list of those classes that satisfy $pm_i \wedge pc_k \neq 0$.
(2) For each of c_j's p.a.d. test if the entry in the timetable for m_i is a single entry and the class is in the list of available classes.
(3) If a class c_f is found that satisfies these requirements then an interchange is possible and two entries are made in the timetable for m_i. The new entry for m_i, c_j, 1, 0 replaces the entry for c_f at the period found in (2).
    The replaced entry corresponding to m_i, c_f, 1, 0 is inserted at the first period found by $pm_i \wedge pc_f$.
(4) Form the new p.a.d. for m_i, c_j, and c_f.
    If the digit d_j represents the period where c_j has been inserted and the digit d_f represents the period where c_f has been inserted, then the new p.a.d. pm_i' and pc_f' are given by

$$pm_i' = pm_i - d_f$$
$$pc_j' = pc_i - d_j$$
$$pc_f' = pc_f + d_f - d_f$$

Logical operations are not necessary since the values of all the digits concerned are known not to cause "carry."

(5) Remove the requirement m_i, c_j, 1, 0 from the requirements list or reduce the number of periods if several were required.

(*c*) In a third attempt if m_i cannot be made available for one of c_j p.a.d. then c_j must be made available for one of m_i p.a.d. This cannot easily be made a direct interchange as in (*b*) since the class timetable does not exist explicitly inside the computer. The program therefore attempts to free c_j by finding the master teaching c_j for one of m_i p.a.d., and at the same time creates a new requirement which it may be possible to insert by (*a*) or (*b*).

(1) Search the timetable for each master in turn to find a master, m_f, that has an entry c_j for one of m_i p.a.d.
(2) If m_f is found then a change can be made and c_j can be inserted in the timetable for m_i at the period found in (1).
(3) Create a new requirement m_f, c_j, 1, 0 for the entry that has been changed and write it into the requirements list.
(4) Form the new p.a.d. for m_i, c_j and m_f. If the digit d_j represents the period where c_j has been entered in the timetable for m_i, and pm_i', pc_j' and pm_f' are the new p.a.d. then

$$pm_i' = pm_i - d_j$$
$$pc_j' = pc_j \text{ (since } c_j \text{ has merely been transfered from } m_f \text{ to } m_i)$$
$$pm_f' = pm_f + d_j.$$

(5) Remove the requirement m_i, c_j, 1, 0 from the list or reduce the number of periods required.

(*d*) When attempts (*b*) and (*c*) have both failed then there is no possibility of fitting the required entry with the current p.a.d. for m_i. Accordingly an attempt is made to change the p.a.d. by moving a class in the timetable for m_i from one period to another. This may then make possible an interchange by attempt (*b*) or (*c*).

(1) Form a list of those classes that satisfy $pm_i \wedge pc_k \neq 0$.
(2) Test if any entry in the timetable for m_i corresponds to an available class c_f.
(3) If c_f is found then the entry corresponding to m_f, c_f, 1, 0 can be made in the timetable at the period specified by $pm_i \wedge pc_f$.
(4) Form the new p.a.d. for m_i and c_f. If d_f represents the period where c_f has been entered in the timetable, d_v represents the period vacated by c_f, and pm_i' and pc_f' are the new p.a.d., then

$$pm_i' = pm_i + d_v - d_f$$
$$pc_j' = pc_f + d_v - d_f.$$

The requirements list is unaltered during attempt (4).

Clearly if (*d*) does not free a period which can be used for an interchange then it will be re-entered to produce a different free period. If no record is kept of the periods that have been freed for a particular master then it would be possible to alternate between two periods merely moving the same class each time. Similarly, in (*c*) it would be possible to move a particular class between the same two masters; for example c_j could shuttle between m_f and m_i. To prevent this happening two records are kept, one of the interchanges made in (*c*) and the other of the digits that are freed for each master in (*d*). Before any change is made these records are interrogated and the cycle is terminated if the change has already been made.

The record for (*c*) consists of c_j, $\min(m_i, m_f)$, $\max(m_i, m_f)$. Keeping the record in this form prevents c_j being transfered back from m_i to m_f. It also has the effect of disallowing any further transfers of c_j from m_f to m_i. This may deny some legitimate transfers, but the possibility of such transfers both arising and being useful did not seem to warrant keeping the complete record showing both the direction of the transfer and the period in the week to which it referred.

Failures

The program can fail to insert an entry if all four attempts are tried and either the cycle is stopped in the last one or no interchange is possible. A failure to insert also occurs if there are no available classes for m_i p.a.d. A stop is encountered as an indication of the failure but on continuing past the stop the failure requirement is rejected and the next insertion is attempted.

Criteria for success

A method for compiling school timetables has been described. Before discussing the application of this method some criteria for judging its success need to be established. There are three main questions that should be asked about any timetable produced.

(1) Are all the master and class requirement satisfied?
(2) Are double periods satisfied without crossing a break?
(3) Are certain subjects (e.g. French or mathematics) acceptably spaced throughout the week?

If the answer to each of these questions is an unequivocal yes, then the timetable produced is completely satisfactory. If, however, the answer to any or all of these queries is no, then the timetable is not perfect, and reasons and remedies must be sought.

The present program answers questions (1) and (2) very simply. For question (2) the answer must always be yes since the program will reject a requirement for a double period rather than permit a break to be violated. The answer to question (1) is also yes unless a failure stop occurs in the program. The third query is not so easily answered. Although the program attempts to space the periods in one subject throughout the week it can fail on two counts. First, if any entries remain from the old timetable no account is taken of these when fitting the new entries. Similarly, when there are insufficient periods on separate days, and two are put on the same day no check is made on the proximity of these entries. Also in the final stage of the program entries are interchanged without regard to the distribution of the periods throughout the week. A little extra programming could add suitable checks.

In order to investigate the relative success of timetables which are not completely satisfactory we can rephrase questions (1) and (3) to read:

(1) What fraction of the master and class requirements are satisfied?
i.e. (The number of master periods satisfied)/(The number of master periods required).

(3) How adequate is the subject spacing?
The inadequacy can be expressed numerically as: (The number of periods on the wrong day)/(The number of master days).

A period is defined as being on the wrong day if it occurs on the same day as another period in the same subject and if there is also a day which contains no periods in this subject. Clearly if there are six periods in one subject and only a five-day week, two periods are bound to occur on the same day and therefore must not count as "wrong" periods. The total number of master days is the product of the number of masters and the number of days per week. Dividing by this quantity serves to standardize the measure of the inadequacy of spacing for different sizes of schools.

Clearly for a completely satisfactory timetable (1) should be 100% and (3) should be zero: any departure from these values gives an indication of the lack of success.

After the timetable is completed these parameters provide a means of measuring its success, but they can give no indication of the causes of the failure, nor can they be used for forecasting the success or failure of a timetable that has yet to be tried. Clearly some parameters of the initial data are needed which are also related to the degree of success of the timetable. From the structure of the method, which depends upon the relative number of periods in sets or double periods and upon the availability of masters, two parameters that would seem worthy of consideration are:

(*a*) (The number of master periods in sets or double periods)/(The total number of master periods available).

(*b*) (The number of master periods required)/(The number of master periods available).

These two parameters reflect most of the more generally accepted constraints of the school. For example the first parameter might indicate the complexity of the sixth form or the number of double periods required in a technical school. Similarly the second parameter defines the master/class ratio or the proportion of free time for masters. Both these parameters are defined in terms of masters rather than classes since the division into classes tends to be artificial especially in the sixth form.

Table 1

Parameters for four schools showing the success and the complexity of the timetable

PARAMETER \ TIMETABLE	1	2	3	4
% periods satisfied	100	100	96	98·6
Inadequacy of subject spacing	0·60	0·24	0·60	0·51
% periods required	80	47·5	86	76
%sets or double periods	22	5	62	46

Examples

Four timetables have been attempted with the present program:

(1) A simulated two-form entry grammar school.
(2) The timetable for a college department.
(3) A one-form entry mixed grammar school.
(4) A five-form entry grammar technical school.

In **Table 1** the parameters are shown for the four timetables. The variations in the values will form the basis for the discussion of each timetable.

The first two timetables attempted did not contain many of the complexities that are met with in real schools. For these timetables all the requirements were satisfied, although the spacing of subjects throughout the week was not completely satisfactory. The two schools whose timetables were constructed differ markedly in size but require timetables with several similarities. Table 1 shows the success and complexity parameters for these schools. Yarm Grammar School, corresponding to timetable 3, is a small coeducational school while South Shields Grammar Technical School, corresponding to timetable 4, is a large boys school. For neither school was it possible to complete the timetable satisfying all the requirements. Although a large school presents difficulties of compilation arising from the scale of the manipulations involved, in this case the smaller one has a much higher complexity parameter showing the number of periods required in sets or double periods. It is this feature that causes most of the compilation difficulties.

The timetable for Yarm G.S. presented two major difficulties. The sixth form were allowed a free choice of subjects which was converted into sets outside the program. This conversion was done by hand and created very complex sixth-form requirements. The school also employed part-time members of staff who were only available on specified days. During these days they were almost fully occupied.

The requirements the program failed to fit into the timetable were mainly concerned with the sixth form or the part-time staff. It was interesting to learn that even when compiling by hand all the requirements could not be satisfied despite an important modification in the sixth-form requirements, which reduced the number of different subject selections from 10 to 8.

South Shields Grammar Technical School for Boys is a large school of approximately 850 pupils. As its name implies, several technical subjects are taught as well as the normal academic subjects. Although this school approached the maximum size that could be tackled by the Pegasus program, with 55 masters compared with the limit of 58, the degree of success did not differ markedly from that of the small school. We drew the tentative conclusion that the size does not directly affect the complexity of the timetable compilation.

The form of the headmaster's specification was much closer to the program requirements than the hand data for Yarm Grammar School. This was probably due to the size of the school, since, for example, in a sixth form of over 150 boys a free choice of subjects could not be allowed. The sixth-form requirements were therefore defined as sets.

During the second stage of the program two sets with four masters, one each with three and two could not be satisfied. All these requirements concern sixth-form classes, which was to be expected since all the sixth-form requirements are sets. In the final stage of the program ten single entries could not be fitted, none of which concerned classes in the first two years. The most obvious cause is the small number of sets required, and a second is the use of the more junior masters who tend not to be involved in the sets associated with the higher forms and who thus have a more flexible timetable.

The method of compiling this timetable by hand differed in one main respect from the computer method. Certain requirements were arranged to take place at fixed times. Thus the majority of the sixth-form time was scheduled in blocks of periods, and all the options for the fifth year were taught at particular times. This part of the timetable was therefore constructed whilst allocating the masters to teach the subjects, and clearly if any clashes appeared then the masters would be interchanged in some way.

A further advantage from this hand method of timetabling is the structure imposed on the timetable at an early stage. It would, of course, be possible to insert the sixth-form and fifth-form timetables as an initial timetable, but this would give a large part of the labour back to the school.

One of the less desirable features of the hand-compiled timetable is the grouping of sixth-form periods in the Arts subjects. This is occasioned by the need to have double and triple periods in science subjects, thus causing the Arts subjects timetabled with them to have the same multiple periods. This restriction was not imposed in the program's requirements since the Arts and Science pupils were in different classes.

Subject selection

The method outlined above might prove adequate for some schools but clearly several points need further consideration. The most interesting of these is the problem of sixth-form subject selection. The two schools whose timetables have been discussed tackled this from opposite points of view. The small school allowed a complete choice of subjects for the sixth form while the large school had fixed sets.

The reason is apparent when the possibilities arising from this free choice are contemplated. For example, if 11 subjects are offered from which 3 must be chosen, the number of possible choices is given by $\binom{11}{3} = 165$. Thus in a sixth form containing 165 pupils it would be possible for each one to choose a different combination. Clearly this number of choices cannot be fitted into a normal week, since it requires that no subject is taught at the same time as any other subject. Thus the amount of time allotted to each subject is restricted to 1/11th of the week.

Maximum number of selections

Suppose that n subjects are offered from which r may be chosen. We are interested in both

(1) the largest number of different selections that can be accommodated under the most favourable circumstances, and

(2) the number of selections that can be guaranteed to fit into the week.

Let $1/s$ be the amount of time allotted to each subject, each week (assuming for simplicity that each subject has the same time, and that s is integral). There are therefore s time segments in the week into which subjects may be put. From the n subjects let k_1 be put in segment 1, k_2 in segment 2 and . . . k_s in segment s.

Let $\phi_r = \{i_1, i_2, \ldots i_r\}$ be any selection of r integers without repetition from 1, 2, . . . s. Then the number f, of different selections of r subjects that is possible is given by $f = \sum_{aU\phi_r} k_{i_1}, k_{i_2}, \ldots k_{i_r}$, and f can be shown to take its maximum value when all the k_i are equal, i.e. $k_i = n/s$. Only integer solutions are possible for the sets problem, and hence two cases arise: if n is exactly divisible by s then $k_1 = k_2 \ldots = k_s = n/s$, and the maximum number of selections is given by $\left(\frac{n}{s}\right)^r \binom{s}{r}$.

If n/s is not integer then s may be divided into u groups having the value k, and $s - u$ groups have the value $k + 1$ where $u \,.\, k + (s - u)(k + 1) = n$. The number of selections of r subjects that can be made is then given by

$$k^r\binom{u}{r} + k^{r-1}(k+1)\binom{u}{r-1}\binom{s-u}{1}$$
$$+ k^{r-2}(k+1)\binom{u}{r-2}\binom{s-u}{2} \ldots + (k+1)^r\binom{s-u}{r}.$$

Table 2 shows values of the maximum possible number of selections for $r = 3$ and a range of values for n and s.

Minimum number of selections

If there are s time segments in the week and n, the number of subjects offered, is greater than s, then it is not possible to accommodate all selections. The number of selections that can be guaranteed to fit into the week irrespective of the subjects offered is very small. For example, suppose seven subjects labelled $A, B \ldots G$ are offered in a week containing four time segments. The arrangement of the subjects in the timetable can be built up by teaching the first subject at time 1, the second subject at time 1 unless the selections forbid this when it must appear at time 2.

Similarly, subject 3 should be taught at time 1 or failing that, time 2 or finally time 3. Clearly it is possible to select groups of subjects that force subject 5 out of the timetable. The minimum number of selections that accomplishes this is one more than the number of selections that can be guaranteed to fit in the timetable.

The following ten selections where two subjects occur in each selection force the fifth subject out of the timetable.

AB forces *B* to appear at time 2.
AB, *BC* force *C* to appear at time 3.
AD, *BD*, *CD*, force *D* to appear at time 4.
AE, *BE*, *CE*, *DE* force *E* out of the four-segment timetable.

If three subjects occur in each selection then the minimum number of selections is only four.

Clearly when searching for the minimum number of selections only $s + 1$ subjects need be considered since any selection made from more than $s + 1$ subjects can be made at least as difficult to accommodate in the timetable by replacing some subjects by others. Accordingly the most unfavourable situation can be constructed from $s + 1$ subjects, and hence the minimum number of selections is independent of n, $(n > s)$ the number of subjects offered.

The minimum number of selections is obtained when the $(s + 1)$st subject is forced out of the timetable. This occurs when all the $\binom{s+1}{2}$ pairs of subjects are present in the selections, and so may not occur together in the same time segment. Since each selection containing r subjects places $\binom{r}{2}$ restrictions on pairs of subjects then a lower bound to the number of selections in the most unfavourable case is given by $\binom{s+1}{2}/\binom{r}{2}$.

This lower bound can be improved slightly by considering each subject in turn. Clearly each subject must occur with every other subject. Now the s remaining subjects can occur in groups of at most $(r - 1)$, thus the minimum number of times each subject must occur is $\left\{\frac{s}{r-1}\right\}$ (where $\{x\}$ denotes the smallest integer $\geqslant x$), which must be integer and therefore exceed $s/(r - 1)$.

Since there are $s + 1$ subjects occurring in groups of r then the above lower bound can be improved to $\left\{\frac{s+1}{r}\left\{\frac{s}{r-1}\right\}\right\}$.

For $r = 2$ this lower bound is clearly attained since it is the maximum number of pairs in $s + 1$ subjects, which is also an upper bound on the number of selections that can be guaranteed.

For $r = 3$ the lower bound has been attained in all the examples tried. For $r > 3$, however, this lower bound cannot always be attained. For example when $r = 4$ and $s = 6$ the lower bound implies that each subject should only occur twice. The subject selections required from the seven subjects labelled *A* to *G* are *ABCD*, *AEFG*, *BEFG*, *CEFG*, *DEFG*, where four subjects only occur twice whilst the remaining three occur four times.

The form of the selections suggests that if s is sufficiently large, i.e. greater than $r(r - 1)/2$, it should be possible to attain the lower bound by avoiding the repetition of subjects within the selections.

Table 3 shows the values found by enumeration of the selections for the minimum number of selections that cause the $(s + 1)$st subject to be forced out of the timetable. The values in italics differ from the lower bound.

The theoretical limitations on a free choice of subject as shown in Table 3 seem severe. In practice, however, because of the tendency of groups of subjects to be taken together the situation is not quite so limiting. For a small school with perhaps ten different subject selections a free choice is a practical possibility. This choice must at some stage be converted into subjects that may take place simultaneously. Accordingly it is worth considering a possible automatic method for converting a free choice into sets.

Table 2

Maximum number of selections from n subjects taken 3 at a time can be fitted into a week containing *s* time segments

	$n = 8$	9	10	11	12	13
$s = 3$	18	27	36	48	64	80
4	32	44	60	81	108	135
5	38	56	80	104	134	171
6	44	63	88	120	160	200
7	50	70	96	129	170	220
.						
.						
.						
n	56	84	120	165	220	286

Conversion into sets

The problem is to find those subjects that may take place at the same time and choose these so that the sets produced will fit into the time available.

If n subjects are offered then an $n \times n$ symmetric matrix A can be formed with elements $a_{ij} = 1$ if subject i and subject j occur in the same selection but zero if not. Clearly any zero in the matrix implies that the corresponding subjects may take place at the same time. Thus if element a_{km} is zero subject k may take place at the same time as subject m, and a reduced matrix can be formed by combining row k with row m and column k with column m.

The elements in the combined row are given by

$$a_{ic} = a_{ik} \vee a_{im}, \quad i \neq k \text{ or } m.$$

Similarly the elements in the combined column are given by

$$a_{cj} = a_{kj} \vee a_{mj}, \quad j \neq k \text{ or } m.$$

The elements when $i = k$ or m and $j = k$ or m are one and are combined on the main diagonal.

The rows to be combined should be chosen so that the maximum reduction in size of the matrix is achieved. The order of the final reduced matrix should not be greater than the number of time segments in the week, since otherwise the subject selections cannot be accommodated.

Extensions

The use of the method outlined above for creating sets from a free choice of subjects brings practically all timetables within the scope of the present program. However, the compilation method requires some modifications to enable all the requirements to be inserted and to space the entries in one subject throughout the week.

An improved spacing of entries can fairly easily be achieved by a more sophisticated method of selecting the required periods from the available ones. Any method of doing this will involve searching the timetable to find entries already made for this subject. The amount of work required should not be significantly increased.

An essential feature that must be incorporated in the program is a method of interchanging set entries. In order to accomplish this economically both the master and the class timetables are required explicitly. Accordingly in the final stage of the program the class timetable must be formed together with its set definitions.

A method of interchanging set entries

An outline is given below of the steps needed to insert a requirement of the form $(m_i, m_j, \ldots)(c_g, c_h, \ldots)$ 1, 0 by interchanging set entries.

(1) Find those periods not occupied with sets which are common to all the required masters.
(2) Similarly find the common available periods for the classes.

Table 3

The number of selections of r subjects that can be guaranteed to fit into a week containing s time segments

	$r = 2$	3	4	5	6
$s = 2$	3	1			
3	6	3	1		
4	10	4	3	1	
5	15	6	3	3	1
6	21	7	5	3	3
7	28	11	6	4	3
8	36	12	8	6	3
9	45	17	9	7	4

(3) Check if any periods are common to both masters and classes. If so insert the set entry, rejecting any single or double period entries, and exit.
(4) For the masters' common periods find the sets in the class timetable that exclude the period for the classes.
(5) Attempt to move each of the obstructing sets in turn to any of the common available digits for the classes. If a move is possible then the set can be inserted by rejecting single periods where necessary.
(6) If step 5 fails attempt steps 4 and 5 with masters and classes interchanged.
(7) If both step 5 and 6 fail, attempt to free some other period by interchanging any set in the master or class timetables. If an interchange is made then steps 1 and 6 can be repeated. It is only after this type of interchange that step 3 might find a common period.

Clearly, interchanging set entries will be a time-consuming operation and should be avoided wherever possible. However, it can be used, if necessary, to insert single-period requirements as well as sets. The single requirement can be treated as a set containing only one master and one class, and can force the movement of a normal set entry.

The method of compilation that emerges from these extensions to the program is to ensure that each entry inserted is in an allowable position. Thus, for example double periods do not overlap a break, nor do more than two entries in one subject occur on the same day. It is more economic in computer time to insert entries in the correct place on the first attempt after a little inspection rather than embark on a long series of interchanges to find a better position.

The degree of success of the trials on real timetable problems gives ground for hope that the method, with the extensions and modifications suggested, will yield complete timetables in most cases. Inevitably, however, there will be some schools whose requirements as initially specified are impossible and others that cannot be solved by this method. For these schools it should be possible to provide a partial timetable which contains all the essential entries but omits those of secondary importance.

Acknowledgements

I am indebted to those headmasters and headmistresses in the North East who have given me copies of their timetables. In particular I must thank Mr. W. E. Egner of South Shields Grammar Technical School, Mr. J. L. Foster of Yarm Grammar School, and Mr. M. Watson of the Newcastle College of Commerce whose timetables are used in this paper.

References

Appleby, J. S., Blake, D. V., and Newman, E. A. (1961). "Techniques for Producing School Timetables on a Computer and their Application to other Scheduling Problems," *The Computer Journal*, Vol. 3, p. 237.

Berghius, J., van der Heiden, A. J., and Bakker, R. (1964). "The Preparation of School Timetables by Electronic Computer," *BIT*, Vol. 4, p. 106.

Csima, J., and Gotlieb, C. C. (1964). "Test on a Computer Method for Constructing School Timetables," *Comm. A.C.M.*, Vol. 7, p. 160.

Gotlieb, C. C. "The Construction of Class-Teacher Timetables," (1962). *Proceedings of the I.F.I.P. Congress*, p. 73, North-Holland Publishing Co., Amsterdam (1963).

Holz, R. E. (1963). "The Gasp Manual," June 1963. Available through the author, R. E. Holz, Associate Registrar, Massachusetts Institute of Technology, Cambridge, Mass.

Lewis, C. F. (1963). *The School Timetable*, Cambridge University Press.

Sherman, G. R. (1958). *The Sequential Method of Scheduling Students*, Computer Research Report, Purdue Research Foundation, September 1958.

THE COMPUTER JOURNAL

Published Quarterly by

The British Computer Society, Finsbury Court, Finsbury Pavement, LONDON, E.C.2, England.

Subscription price per volume £3 10s. (U.S. $10). Single Copies £1.

All inquiries should be sent to the Assistant Secretary at the above address.