The Ontario school scheduling program

By J. Lions*

The method described by Csima and Gotlieb has been successfully implemented in a computer program for the construction of high school timetables. A production-oriented computer program has been written with features for handling special requirements, and lunch and room assignments. Results of applying the program to real school data are discussed.

That the application of digital computers for the production of school timetables might offer substantial advantages has been well known for some time. However, early efforts to realize these advantages wherein the human was replaced by a computer without any fundamental change of procedure, proved not to be very satisfactory. It appears that in this, as with many other computer applications, the effective use of the new tool, the computer, has necessitated the development of fresh approaches and techniques. In the present connection, a new approach was suggested by Gotlieb (1962), and subsequently a revised and more complete statement of the method was given by Csima and Gotlieb (1964).

After a large number of tests on problems of modest size had been successfully completed (Duncan, 1965), the method was considered promising enough to warrant testing with real school problems. Since this was a major programming task, it was no longer suitable for a university department, and so the task was undertaken in May, 1964 by J. Kates and Associates on behalf of The Ontario Institute for Studies in Education (then the Department of Educational Research, The Ontario College of Education).

At the outset, there was a serious difficulty to overcome. With the method as originally described, the computation time to solve one problem grows exponentially with the size of the problem and is quite astronomical for the problems posed by the largest schools in Ontario. Necessary modifications to overcome this and to cut the computation time down to size were discovered and implemented early in 1965.

The basis for the method is essentially empirical, and the first purpose of the Mark I version of The Ontario School Scheduling Program (OSSP) was to establish this basis for problems very much larger than those for nine teachers and nine classes which has been examined extensively initially. In this it was successful, and before its development was discontinued in September, 1965, the Mark I version was used to produce a timetable for the Queensway Senior Public School in Etobicoke, Ontario, which, with some modifications, was used during the 1965–1966 school year.

The Mark I version was discontinued in favour of the development of a production-oriented, Mark II version.

The latter allows a more convenient form for the preparation of data, makes better use of the computer, handles the special requirements much more effectively and produces a more comprehensive listing of the final timetable than the Mark I version. During the middle of 1966, extensive experiments with the Mark II program were undertaken with data supplied by eight schools. Although some failures were recorded, the results were generally encouraging, and two of the schools will be using computer produced timetables during the year 1966–1967. The success achieved is most probably best measured by the fact that the group of schools who have participated so far have expressed the wish that the project be continued and expanded in the future.

The Mark II version is described below. Discussions of the basic algorithm, of the problem for a single day, of the problem for a whole week, of some technical details and of the results obtained so far are given in that order.

The basic model

The theoretical basis for the program is the method described by Gotlieb (1962) and Csima and Gotlieb (1964), but with the "tight set search" replaced by an equivalent but more efficient algorithm based on the Hungarian Method (Lions, 1966a). The method assumes a model of a school which is described below.

A school is considered to consist of $N$ classes and $N$ teachers, who meet together during the $L$ hours of the school day. (Since real schools usually have more teachers than classes, pseudo-classes and pseudo-teachers are included to even up the numbers). During each hour of the day, each class is to meet exactly one teacher, and each teacher exactly one class. During the whole day, each class is to meet a total of $L$ teachers (not necessarily all different) and each teacher $L$ classes. The number of hours $R_{ij}$ during the day that class $i$ is to meet teacher $j$ is called the requirement for class $i$ and teacher $j$. The matrix $(R_{ij})$ is called the requirements matrix and satisfies consistency conditions that the requirements for each class and for each teacher must sum to $L$. From a theorem of Konig (1916) it can be deduced that any consistent requirements matrix whose row and column sums are each $L$, can be expressed as the sum of $L$ permutation matrices, and hence that any timetable problem for which the requirements matrix represents the only imposed conditions, must always have a solution.

* 20 Spadina Road, Toronto 4, Ontario, Canada.
School scheduling

From among these opportunities, it may be possible to choose a set of $N$ opportunities involving each teacher and each class exactly once. This set then defines a possible schedule for the hour. Obviously if a point were ever reached in the calculation where no schedule could be constructed for an hour, then it would not be possible to construct a complete timetable.

Usually it is possible to construct several schedules for the hour (one of which is destined to be in the finished timetable). Now it can happen that none of these schedules uses a particular opportunity and, therefore, this opportunity must be discarded since it corresponds to an event which cannot occur in the finished timetable. Once an opportunity is discarded, then the number of schedules that can be constructed for some class and for some teacher is reduced, with the result that additional opportunities may also have to be discarded. Schedules for classes and teachers are examined in the same way as for the hours. The process of discarding unusable opportunities, or array reduction, may cause a chain reaction affecting several classes, teachers and hours. Once the array reduction has been completed, if there still exists at least one possible schedule for each class, for each teacher and for each hour, then the acceptability test is said to be successful; otherwise it is unsuccessful.

The acceptability test is based on the implications of a set of conditions which are obviously necessary if a timetable is to be constructed. Csima and Gotlieb (1964) went further to postulate that these conditions were also sufficient (and so have termed the test a feasibility test). Subsequent investigations (Duncan, 1965) tended to confirm this. The $3 \times 3 \times 3$ counter-example given by Csima and Gotlieb (1964) was dismissed by them as inapplicable, since some of the opportunities had been arbitrarily eliminated and, if restored, would not be eliminated again by the array reduction process. However, after constructing several hundred timetables (derived from fourteen different data sets), we are now able to assert that instances of failure of the method (i.e. the incorrect acceptance of one or more preassignments) do exist (Lions, 1966b), but seem to be quite rare. Because the probability of completing a timetable at each attempt is so high, the method does form the basis of a reliable procedure, and any instances of failure can be side-stepped relatively easily by reworking the problem with small perturbations.

The daily problem

The daily problem is to find hours for each teacher to meet his classes, and each class to meet its teachers, without conflicts or inconsistencies, during a single day. Real schools have requirements of many varied types which they wish to see embodied in their timetables. OSSP is designed to handle requirements in the range covered by the following types:

(1) a teacher and class are to meet for a certain number of periods during the day;
(2) a teacher (class) is without obligation to meet any
School scheduling

class (teacher) for a certain number of periods during the day;

(3) a teacher and class are to meet at a certain hour of the day;

(4) a teacher and class are to meet for two or three periods in succession during the day;

(5) a teacher and class are to have the exclusive use of a certain room at the time they meet;

(6) Teacher $T_1$ is to meet class $C_1$ at the same time as Teacher $T_2$ who is to meet class $C_2$ at the same time as $\ldots T_m$ who is to meet class $C_m$.

(7) Each of the group of teachers $T_1 \ldots T_m$ is to meet a group of students drawn from the classes $C_1 \ldots C_m$ for a certain number of periods in the day.

Types (1), (2) above represent normal requirements. If all the requirements are normal, the timetable problem reduces to the simple problem mentioned at the beginning of the last section. The examples (3)–(7) represent special requirements. Using (4) provides some flexibility in the lengths of teaching sessions. The use of (6) or (7) allows students to choose among optional courses. The use of (5) reflects the schools' desire to concentrate specialized teaching facilities in certain areas.

Requirements of type (2) invariably appear when there are more teachers than classes in a school. In order that the consistency conditions should continue to hold, pseudo-classes may be introduced so that when a real teacher is not meeting a real class, he is meeting a pseudo-class. It turns out to be most convenient if each teacher is assigned his own pseudo-class which he meets whenever he does not meet a real class. A symmetrical arrangement between real classes and pseudo-teachers is also set up. Each class meets its pseudo-teacher at least once per day for the subject "lunch".

For requirements of type (6), the teachers and classes are naturally grouped in pairs, and a requirement for each pair is entered into the requirements matrix. Requirements of type (7) can be treated similarly, except that the pairing of teachers and classes may be somewhat arbitrary. If there is an excess of teachers over classes spelled out in the requirement, some of the teachers are paired with their pseudo-classes. (A similar situation holds if there is an excess of classes.)

To process a single daily program, the program scans the set of requirements, and builds up in the computer core memory a representation of the requirements matrix and a list of the special requirements for the day. The special requirements are then scanned sequentially in an order based on their stated priorities and estimates of their relative difficulties. For each special requirement a set of preassignments is generated which will embody the structural requirements and be compatible with the preassignments already accepted. Should the acceptability test subsequently fail, the set of preassignments is rejected and a new set is selected. If suitable sets of preassignments do not exist or are all unacceptable for a special requirement, then its structure is modified or in the last resort, abandoned.

When the special requirements have been assigned, a lunch assignment phase is entered. Certain periods of the day are designated as "lunch periods". During the assignment of the special requirements, care is taken to ensure that each teacher and class will be able to take lunch during at least one of the designated periods. Where classes have a choice of lunch period, the lunch assignment program resolves these choices so that the capacity of the school cafeteria will not be exceeded at any time, while endeavouring to give each class the same lunch period as on the previous day.

After the special requirements and lunch have been dealt with, the normal assignment phase is entered. During this phase, the timetable is methodically completed, hour by hour, by making arbitrary choices whenever two or more alternative assignments for a class at the hour may be made. The number of such choices which need to be made turns out to be quite small. Thus, in one example recently analysed, where 301 assignments had to be made, 128 has been made by the end of the lunch assignment phase. Of the remaining 173, there was a choice for 53, and 120 were forced.

The main difficulties with the daily problems centre around the assignment of the special requirements. Firstly, schools are tempted to issue more special requirements than they could normally hope to satisfy in a manually constructed timetable. This is human nature and is not serious as long as it is agreed which special requirements are essential and which are merely highly desirable. More serious, however, is the problem which sometimes arises when it is impossible to fit all of a certain group of essential requirements into the timetable for a single day. This problem is often compounded because the reasoning needed to show impossibility may be quite obscure. The following example is derived from an actual set of requirements supplied by a school: A set of requirements including requests for pairs drawn from a set of circumstances $A$, $B$, and $C$ in the following combinations,

$$(A \text{ and } B) \text{ three times}$$
$$(B \text{ and } C) \text{ three times}$$
$$(C \text{ and } A) \text{ three times}$$

cannot all be satisfied during an eight hour day, although each of $A$, $B$, and $C$ is requested only six times. The reason is that no two pairs may be assigned at the same hour without conflict, and hence a minimum of nine hours is needed to assign all the pairs.

Checking procedures to catch this sort of essential infeasibility will eventually form an important part of the program, since the algorithm is not "guaranteed" (i.e. it is often possible to improve on a first timetable by judicious modification of the data structure and re-running the job) and it is most desirable to avoid wasting analyst and computer time.

Another problem in the making of sets of preassignments for the earlier special requirements is that of anticipating the needs of the later requirements. (In this respect, the acceptability test attempts to protect the
needs of all the normal requirements.) A “look-ahead” procedure has been implemented which “advises” the program to choose sets of preassignments for the special requirements which will avoid blocking out the later special requirements. For example, if special requirement 20, which conflicts with (i.e. cannot be assigned to the same hour as) special requirement 45, can be assigned to either hour one or hour two while requirement 45 can only be assigned at hour two, then the program will choose to assign requirement 20 at hour one. There is, however, no guarantee that when requirement 45’s turn for assignment comes around it will still be possible to assign it to hour two.

By the end of the normal assignment phase, the schedule for meetings between teachers and classes has been decided, and some rooms have been assigned to teachers during the processing of the special requirements. In the final phase before the timetable is recorded, the remaining rooms are assigned to the teachers on the basis of the teachers’ expressed preferences. (Teachers may give ranks between 1 and 10 for up to 10 rooms. All preferences of rank 1 are considered before those of rank 2, and so on.) There seems to be some difficulty to devising a room assignment algorithm that is widely acceptable. The layout of the school and the school principal’s own preferences are often important factors which are difficult to describe. The present rules seem to make satisfactory room assignments about 90% of the time, and facilities will eventually be provided for adjustments to the room assignments and other details to be made before the final version of the timetable is printed.

The weekly problem

Schools normally provide their timetable requirements formulated as requirements for one week. In the present context the weekly problem is interpreted as the problem of dividing the set of weekly requirements into five or six subsets, each of which then defines one daily problem. It has already been pointed out by Gottlieb (1962) that there is a formal similarity between the weekly and daily problems. The former involves taking a set of requirements and dividing them into groups by days, while the latter involves dividing a set of requirements into groups by hours. Experience shows that the weekly problem is much the easier of the two problems—a teacher or class may take part in several activities during a single day but only in one activity during a single hour—and a solution procedure for the weekly problem can be considerably less complex than for the daily problem.

The decision to divide the whole problem into a number of subproblems—the weekly problem and the daily problems—was occasioned by the limitations of the available computers. The amount of computation involved in one application of the acceptability test is considerable, and for a large school, one daily problem alone taxes the power and storage of even a large present-day computer. From this necessity also comes one virtue. Provided the weekly requirements are distributed “smoothly” among the daily problems, certain hard-to-

define aesthetic requirements for the finished timetable can be satisfied to a large extent. (For example, if a class and teacher are to meet three times per week, then “once on each of Monday, Wednesday and Friday” is considered very good, “once on Tuesday, Thursday and Friday” is quite acceptable, and “twice on Thursday and once on Friday” is very bad.)

The method used in the program for the weekly problem is quite empirical. Requirements are considered one by one in order of their stated priority and complexity. All possible breakdowns of the requirement among the days of the week are considered, and that chosen which has the highest figure of merit. The figure of merit is based partly on the distribution of the requirement and partly on the residual availability of teachers, classes and rooms among the various days. A simple “look-ahead” procedure, using a tight set search is provided to avoid situations at the tail end of the list of requirements where, for example, in a five-day timetable, class A and teacher B are required to meet but class A is completely committed on Monday, Wednesday and Friday, and teacher B is completely committed on Tuesday and Thursday.

The schools prepare their input data on self-coding forms which each school draws up for itself. The requirements are defined by marks placed next to the names of the appropriate teachers, classes, subjects and rooms on the forms. The forms contain provision for entering the number of periods needed, including doubles and triples. They also contain small grids with a cell for each period in the week; these grids can be used to note information regarding preassignments, or assignments to be made during a restricted part of the week. A normal requirement is entered on a single sheet, but some special requirements need several sheets. Each requirement is given a separate number and a priority.

In the first phase of the program, the requirements are read in, checked for internal consistency, and sorted into an order based on the stated priority and their complexity. The requirements, in their new order, are then split, one by one, into parts—one for each day of the week (some of the parts may be null). Checks for duplicate pre-assignments and correct total requirements for each class, teacher and room are made.

For the initial runs the program is set up to terminate before the time-consuming processing of the daily problems is begun. A report is produced which displays all the input data in a form suitable for visual checking, and which displays also the division of each requirement by day, the so-called “Assignment-to-Days”. If the “Assignment-to-Days” is not satisfactory, the data can be modified and the program rerun. However, it has proved possible and effective to develop improvements to the Assignment-to-Days by hand (taking of the order of one or two hours) and to incorporate these in the computer produced Assignment-to-Days on later runs. This intervention by hand into the computation is consistent with the philosophy that the computer is a tool rather than a completely automatic servant. It also has the positive virtue of causing the schools to check the intermediate
computer output thoroughly, as there are many possible data errors which can go undetected by the computer.

After a satisfactory "Assignment-to-Days" has been obtained, the program is allowed to go on, to complete each of the daily problems and to enter the final stage which gathers the outputs for each day and produces printed schedules for each class, teacher and room.

**Program details**

The program, in its present form, consists of some 7,000 FORTRAN IV statements and about 300 statements in MAP, and is implemented on the IBM 7094 at the University of Toronto which has a 32K core memory of 36-bit words. Three magnetic-tape drives are used for the intermediate storage of data during the calculation. The program is arranged in twelve links, and core storage requirements in the tightest situation is approximately 9L words for each teacher and class, where L is the number of hours in the school day. About 10% of the storage is manipulated at the bit level, using a formula 3L bits per word. The MAP subroutines are utilized to implement logical operations not directly accessible in FORTRAN for the bit manipulations. The innermost subroutine for the array reduction has been recoded from FORTRAN into MAP, improving the execution time by a factor of about 3. The data arrays for the time-consuming daily problem are contained entirely in core. The maximum size of school which can be handled depends on a number of considerations, and is between 50 and 60 classes, and 70 and 80 teachers when a nine-hour day is being worked.

**Results**

A series of experiments was carried out during the summer of 1966 with the Mark II program, and with current data supplied by eight different schools. The schools ranged widely in size and in the variety and complexity of their problems. In Table 1 the schools, each identified by a three-letter code, are listed in an approximate order of difficulty. For each school almost two parameters: the number of classes (NC), the number of teachers (NT), the fraction of required teacher periods involved in special requirements rather than normal requirements (B1), and the fraction of the total available teacher periods which are considered (B2). The parameters B1, B2 are measures of complexity and difficulty, and were used by Barraclough (1965). (The definition of B1 used here apparently differs slightly from Barraclough's definition in that there preassigned periods are classed with "sets" and doubles in computing B1.)

Some individual comments on the schools are in order. The full school, VCP, is too large for the available computer, and the problem which was considered concerned only the vocational part of the school. This forms the kernel of the main timetable and has rather unusual properties, as many of the classes and teachers have considerable slack. The main difficulty was to schedule double and triple periods into a number of heavily used special-purpose teaching areas. For WNA, a very high number of preassignments caused a very high value for the parameter B1. For the very small school HCT, somewhat unusual conditions removed all choice from the assignment of the teacher spare periods and resulted in a very tight problem.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>List of schools</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCHOOL</td>
<td>NC</td>
</tr>
<tr>
<td>----------</td>
<td>----</td>
</tr>
<tr>
<td>1</td>
<td>HBD</td>
</tr>
<tr>
<td>2</td>
<td>NBK</td>
</tr>
<tr>
<td>3</td>
<td>WNA</td>
</tr>
<tr>
<td>4</td>
<td>SJA</td>
</tr>
<tr>
<td>5</td>
<td>HCT</td>
</tr>
<tr>
<td>6</td>
<td>VCP</td>
</tr>
<tr>
<td>7</td>
<td>SWL</td>
</tr>
<tr>
<td>8</td>
<td>PCT</td>
</tr>
</tbody>
</table>

Except for HCT and VCP, the parameter B2 does not differentiate between the schools. This is to be expected as the amount of free time per teacher is a policy decision not likely to vary much between schools. If WNA and VCP are ignored, then the parameter B1 divides the schools into two distinct groups which correspond roughly to "easy" and "difficult". The school HCT would have been easy except for the high value of the parameter B2. Thus, to some extent, B1 and B2 do measure the degree of difficulty in a timetable. However, they do not distinguish at all between the last two schools, SWL and PCT, where, in fact, the latter is more difficult by a considerable amount. All schools, with one exception, worked a five-day week with nine periods per day including lunch. SJA works a six-day cycle with eight periods per day.

**Data collection**

The data was prepared by each school ready for key-punching. Although a considerable volume of paper is produced, the schools have found the method convenient to use and this form of input virtually eliminates transcription and spelling errors. However, a number of problem areas were discovered. Those due to misunderstandings were dealt with relatively easily, but others are more significant. Consider the following requests:

"9A, 9B, 10B take Geography, Latin and French for 5, 6 and 7 periods per week respectively; The five periods of 9A Geography are to coincide with five out of the six periods of 9B Latin; In turn the six periods of 9B Latin are to coincide with six out of seven periods of 10B French."

or

"Students in 9C are to be offered the opportunity to take German with the class 10C. Students in 9C taking this option must give up one period each of French, Science and Mathematics and two periods of Physical Education."
Various ways of coping with these within the present framework can be devised but they are clumsy and awkward and they involve the introduction of unnecessary additional restrictions. The largest school attempted, PCT, has three streams: academic, commercial and vocational, but it is rather small for such a school. In order to make effective use of the facilities and teaching staff, the principal has been accustomed to “weaving” the various courses very closely together. The result was that, although considerable effort was expended, it was not possible to obtain a data set which adequately defined the school’s timetable problem! One reason is that the basic supposition inherent in the setting up of a requirements matrix, namely that the total requirement can be expressed as a set of independent, simple requirements, breaks down. Basically the timetable problem is not well-structured even though, for many schools, the kind of structure suggested by Gottlieb (1962) is not too restrictive.

**Division of the weekly problem into daily problems**

The heuristic algorithm for carrying out the division of a weekly problem into sets of daily problems has proved to be adequate for finding a feasible division and in obtaining reasonable distributions of the periods of each subject. Even so, it has proved to be quite easy to generate changes by hand to improve the distribution of teacher spare periods and to avoid the kind of problem mentioned by Barraclough (1965) of subjects assigned “on the wrong day”. In fact it has been possible to go further and, for example, for a three-period subject to eliminate, to a very large extent, distributions of the kind “Monday, Tuesday, Wednesday” (three successive days) in favour of distributions of the kind “Monday, Tuesday, Thursday” (no more than two days in succession). For two schools, HCT and VCP, the initial division of the weekly problem created some infeasible daily problems and had to be reworked.

**Solution of the daily problems**

The basic algorithm virtually assures that the timetable will be completed once the special requirements have been assigned, and hence the interest in the daily problems centres mainly on the assignment of the special requirements. As pointed out above, the program uses “look-ahead” procedures to anticipate the needs of requirements to be assigned after the current one. For the easier schools (HBD, NBK and WNA), the “look-ahead” procedures proved to be quite adequate. For the remaining schools, an average of three special requirements were dropped on the first run for a single day. When special requirements were dropped, the diagnostic output produced by the computer was analysed and the job was rerun after certain analyst controls were set to control the assignment of some requirements and/or to modify the order in which the requirements were processed. Very often, as many as six runs have been needed to obtain a satisfactory solution for a single daily problem, with the average number of runs being about two or three. Frequently analyst intervention to correct one problem was frustrated by the emergence of a new problem. The law of diminishing returns definitely applies to the “analyst intervention-rerun” cycle and there occurs a point at which it is better to turn the problem over to the school for the “final touches”.

**Time used**

The daily problems consume by far the largest part of the total computer time used. Times to solve a single daily problem for each school (except SJA) are plotted logarithmically against school size (sum of classes and teachers) in Fig. 1. (The times were obtained with the IBM 7094-II at the University of Toronto.) Considerable variation will be noted. Certain options available with the look-ahead procedure can increase the time by as much as a factor of three, and variation between the days of the week and different runs for the same day can be as much as 25%-50%. The temptation to draw a regression line in Fig. 1 with a slope between 1·5 and 2 has been resisted as the differences between schools introduce considerable bias. In particular for the smallest school, HCT, a major component of the reported time is associated with tape movements for the loading of successive program links.
Final results

A complete timetable was obtained for HBD, and timetables with only one missing assignment were obtained for each of HCT and SWL. For NBK, a timetable which lacked five assignments to a heavily used room, was created without any analyst intervention, and the experiment was not continued further. Experiments with WNA and VCP were not carried as far as possible, as needed advice could not be obtained from the school principals who were on vacation. Two timetables were constructed for SJIA, at the beginning and end of the experiments. The first timetable was completed by hand by the principal in about one day and is now in use. The second was better and only required about three hours manual work to correct some missing room assignments.

The major disadvantage of the present program is the number of computer runs needed to reach a satisfactory point. In particular, it is not the total computer time but the total time which elapses between the initial data submission and the creation of a final timetable which is the major problem. Even for a school as large as SWL, the total IBM 7094 computer time needed for all runs was less than three hours and is not considered excessive at this stage. Improvements to the program which will reduce the number of runs needed are now under consideration.

Concluding remarks

The experiments had to explore, as one of their aims, the range of different problems which exist. Although the types of individual requirements to be encountered were known, the quantities and combinations in which these would occur, were not known. It was found that schools do differ greatly in the emphasis which they give to different aspects of the timetable, and items which some schools considered very important were not even mentioned by others. Traditionally the timetable has been a vehicle for a principal to express much of his own philosophy on school organization, and the significance of this should not be underrated.

Any school timetable is necessarily a compromise. In particular, it is usually impossible to incorporate in a single timetable all the requirements a school could conceivably wish for, and one school principal has even said that he finds it impossible for him to express all the requirements needed for his school. Very roughly, it is possible to divide the requirements into two groups: "formal" and "aesthetic". The formal requirements are capable of being stated in a fairly unambiguous manner, relate to certain structures which must be built into the timetable (double periods, parallel classes, etc.), and possibly restrict the structures to certain predetermined parts of the week. Aesthetic requirements are entirely concerned with the relationships between assignments and their "neighbours". They usually involve value judgements and are difficult to categorize.

A large scale of formal requirements leaves comparatively little scope for the satisfaction of aesthetic requirements. The reverse is also true. Thus, the severest criticism has come from a school for which the formal requirements presented no difficulty at all! The assessment of quality is both a matter of practical consequence and a difficult problem.

Although objective criteria to assess the quality of a finished timetable can be developed (as has been done by Barracough (1965)), the most important criterion is essentially subjective and is inherent in the question: "Will the school buy it?" A timetable produced very early in the experiments, was definitely not perfect but it was taken by the school principal, and converted into the timetable which his school is using during the 1966–1967 academic year. In this case, the computer results were considered to be very valuable. On the other hand, a timetable for another school and which was perfect on the basis of the criteria suggested by Barracough, was declared inadequate. It is not irrelevant that these two judgements were made by two different persons. Even so, it is clear that the success of a computer-based system for producing school timetables cannot be assessed from objective criteria alone.

With this reservation, it is the conclusion of this report that substantial progress has now been made towards the setting up of a computer-based timetabling system for schools in Ontario.

Acknowledgements

The development of this project has been supported by The Ontario Institute for Studies in Education, and the permission of the Director, Dr. R. W. B. Jackson, to publish this paper is gratefully acknowledged. The project owes a great deal to the direction and guidance provided by Dr. B. A. Griffith, and important contributions to the project have been made by Messrs. D. Kaplan, M. Nahon and P. Pedersen, and Dr. M. A. H. Dempster. The co-operation of Mr. J. Anderson of the Scarborough Board of Education, Dr. E. N. McKeeown of the Toronto Board of Education and of the several school principals who have supported the project by their interest and assistance is also gratefully acknowledged. Thanks are due to Dr. C. C. Gotlieb for the use of the IBM 7094 computer at the Institute of Computer Science of the University of Toronto.

The Ontario School Scheduling Program is the property of the Government of the Province of Ontario and may be made available to interested parties at a nominal cost.

References


20
Book Review

Technical Information Center Administration, 2 Vols., edited by A. W. Elias, 1966; 171 and 169 pages. (London: Macmillan, 50s. and 52s. 6d.)

These are proceedings of two practical conferences held at the Drexel Institute of Technology, and the first volume for 1964 focuses attention primarily on the basic skills and services of technical information services, reporting on abstracting, public relations, staff administration, and related subjects. Of the 13 papers, only the two covering indexing and abstracting control (from representatives of the Auerbach Corporation), and one from Benjamin Cheduleur on computer management have really any significance for review in this Journal. One possible addition is the discussion by Alec Peters of the Franklin Institute. This paper covers cost control of information work, and it is an interesting application of computers; and although on the face of it this may appear to be a humdrum computer application, it is basically conventional cost accounting is probably unknown in libraries.

Benjamin Cheduleur presents the sort of paper that all English librarians should be thinking towards—the problems that the librarian will face as data processing managers organizing a computer centre, its personnel, its systems, and its software; not really information retrieval, but one of the few papers anywhere that introduces the computer to librarians as librarians.

The second volume includes such fields as technical writing, internal publications, translating services, and user education, together with another of those necessary warning papers. E. H. Brenner discusses the control of quality of information systems which must be maintained over quantity, and terms this type of choice as “equivocation”—which seems fair enough. This particular paper would have been very relevant to the theme of the Newman collection on compatibility to be reviewed.

Alan Rees contributes a very useful evaluation of evaluation methods, which notes the literature and covers problems of relevance and the weaknesses of design experiments.

The most pleasant discovery is the contributor who notes the countless economical and trouble-free systems that have functioned without failure for years. He is reporting from his own field of literature, and his paper is called “The giant mechanical brain in the cut-out skull”. He is none other than Frederik Pohl, and part of his contribution is the idea that information should be made to communicate itself. I think for once that reality has beaten science-fiction to the draw, because this is what an SDI scheme is ideally designed to do, and this is why librarianship should be an active profession, to give the systems which can provide Pohl with the information that he did not know he wanted.

Neither of these volumes is about the meta-methodology of information retrieval; they are both about the problems of running libraries and providing service. That far they are of some relevance, but they represent nothing new. TICA seems, however, to be a practical conference and one can imagine each of these papers generating much discussion. These proceedings could stand perhaps as transatlantic supplements to the ASLIB Handebok of special librarianship, with or without Frederik Pohl.

R. D. GEE