AN ABSTRACT OF THE THESIS OF

HARVEY ANDREW THOENNES (Name) for the MASTER OF SCIENCE (Degree)

in MATHEMATICS (Computer Science) (Major) presented on July 14, 1972 (Date)

Title: STUDENT SCHEDULING BY COMPUTER: A DISCUSSION OF AN ALGORITHM BASED ON COURSE REQUESTS AND PRIORITIES

Abstract approved: Redacted for Privacy

Harry Goheen

A solution to the problems of computer scheduling of students based on course requests rather than section requests is discussed. The solution involves keeping all course data in core during the scheduling process to enable the scheduling program to run as fast as possible and using a scheduling algorithm which tries all possible combinations of sections for the requested courses until a conflict-free schedule is found, or until it is determined that the remaining conflicts are unresolvable. The algorithm schedules recitations and labs where required and maintains section balancing in terms of the percentage of scheduled stations for lecture, lab, and recitation sections within a course. Several methods of giving the student more choice in the sections he will be scheduled for are discussed. These include instructor preference and course request alternates. These features are not implemented in the algorithm currently being used.

Student Scheduling by Computer: A Discussion of an Algorithm Based on Course Requests and Priorities

by

Harvey Andrew Thoennes

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1973

APPROVED:

Redacted for Privacy

Professor of Mathematics
in charge of major

Redacted for Privacy

Acting Chairmen of Mathematics Department

Redacted for Privacy

Dean of Graduate School

Date thesis is presented July 14, 1972

Typed by Mary Jo Stratton for Harvey Andrew Thoennes

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

STUDENT SCHEDULING BY COMPUTER: A DISCUSSION OF AN ALGORITHM BASED ON COURSE REQUESTS AND PRIORITIES

I. INTRODUCTION

Background of Program Development

The scheduling programs at Oregon State University were developed as part of a registration system designed to take packets of requests for courses and services submitted by students as input and produce schedules, bills, and class lists as output, as well as provide data for enrollment summaries and other reports.

The registration process as a whole was required to implement administration policies which specified that a) for normal registration students would be scheduled in the order they submitted their packets; b) for pre-registration they would be scheduled in packet-submission order within a specified alphabetic sequence; c) course requests would specify the course only, not the section desired; d) schedules would be printed in alphabetic order for distribution; e) students would be able to request services such as a yearbook, insurance, and parking permits; f) students would be able to specify courses they wanted graded on a satisfactory/unsatisfactory (S/U) basis; and g) free time requests would not be submitted through the packet but would be submitted to the Registrar's Office.

Additional policies applied directly to the scheduling algorithm to

be developed. These were a) courses would be scheduled in the order of the request cards in the packet; b) in case of unresolvable conflicts between courses, the first course requested would be scheduled; c) labs and recitations would be scheduled by the program without specific requests for them; d) an upper limit (19 credit hours) for one student would be imposed during the scheduling runs; e) at most only one Physical Education (PE) activity course would be scheduled; f) section balancing in terms of percentage of stations left would be maintained for lecture, recitation, and lab sections within each course; g) free time requests would be honored unless the only available sections of a course conflicted with them; h) unless over-ridden by a card in the packet, each student would have at least a half-hour free for lunch each day during the 11:00 to 14:00 time period, and i) 100,000 (originally 5,000) attempts would be made to give a student a conflict free schedule.

The Registrar's Office also wanted some method of handling instructor preferences by the students but this was felt to be infeasible for the Fall 1969 registration deadline. It was decided that this feature would be implemented as soon as a feasible method could be developed. This feature is now in the final stages of checkout and will be implemented Fall Term 1972.

Besides administration policies, other major constraints on the design of the registration system as a whole and therefore the

scheduling subsystem in particular were the time frame planned for the Fall 1969 registration, and the amount of core storage available on the Control Data Corporation (CDC) 3300 computer under the Oregon State Open Shop Operating System (OS3). The time frame called for generating schedules from packets overnight. This constraint made it mandatory that the registration subsystems run as fast and as efficiently as possible. Under OS3 each user has a maximum of 64K (65, 536) 24 bit words available for use. This constraint placed a limit on the amount of course data that could be held in core at one time.

Since the scheduling subsystem was the one area where one or more reruns might be desirable, it was especially important to develop a fast subsystem. It was decided that this could be accomplished by keeping the scheduling data files at a minimum size, and by keeping all course data in core during the scheduling process.

On the basis of the above policies and system constraints, the registration system was designed to have the following operational stages:

1) Packets are processed and edited to generate a course requests file with free time requests merged in.
2) Students are scheduled.
3) Scheduled students' records are sorted to alphabetic order.
4) Scheduled students' records are decoded and additional

section data including course title, instructor, building and room is added.

5) Students are billed and schedule/bills and class lists are printed.

These stages satisfied the policy requirements and the programs in each subsystem area would fit in the core available.

Once the subsystems to be developed were defined, the next decision to be made concerned the programming languages to be used. It was decided to use FORTRAN for the basic programs for the following reasons:

1) FORTRAN would be more easily converted to another machine or system if necessary or desirable.

2) Programming and testing could be done faster.

3) It would be easier for other programmers to work on the system.

4) Perhaps most importantly, it was felt that using COMPASS, the assembly language for the CDC 3300, would not significantly improve wall clock time for the processing runs.

However, COMPASS subroutines would be used to increase speed where FORTRAN would not be efficient, and to access arrays which could not be referenced by FORTRAN. Of the 64K words available to a user under OS3, only 32K is accessible by FORTRAN programs. The other 32 K can only be accessed by COMPASS routines.

Development Stages of the Scheduling System

The development of the scheduling subsystem has gone through three distinct stages. The first stage consisted of development of the original programs used for Fall 1969 registration. The second stage involved rewriting the original programs to eliminate the problems which arose during the Fall 1969 scheduling run. The programs developed during this stage were used Winter 1970 through Fall 1970. The third stage of development consisted principally of improving the scheduling algorithm to make it run faster. The current programs are the result of these modifications and they have been used from Winter 1971 through Spring 1972.

The remainder of this chapter will indicate the problems which arose during the initial run and the modifications made during the second and third stages of development.

The basic problems which appeared during the Fall 1969 scheduling run were:

1) The scheduling program took too long to run (approximately 10 hours).
2) Sections were not balanced.
3) The wrong time offered data was printed on some schedules.
4) Some specific courses with recitations and labs were scheduled incorrectly.

5) Some courses had more than 100% of the stations scheduled.

6) The school or class restrictions did not work correctly in some courses.

7) Some students did not get a PE activity course or they were scheduled for the wrong one.

There was no record kept of the number of occurrences of each of these problems but an indication of their frequency is the fact that there were approximately 18,000 drop/adds that term compared to approximately 16,000 each term since then.

To eliminate these errors and to introduce more flexible school, class, and sex restrictions, the course data structure in core, and therefore the scheduling algorithm, was changed. These modifications were the objective of the second development phase.

The changes made principally affected courses having recitation and/or lab sections. For these courses, instead of keeping the time and restriction data as they were specified for each section individually, time and restriction data were generated for every valid combination of the lecture, recitation, and/or lab sections for a specific course. This change made it possible to schedule a course, checking the time and restrictions for the combination, rather than checking the time and restrictions for each section separately. Since there was one course which had over 25,000 valid combinations, the

original approach was retained as a special case in the current scheduling program. This was generalized by treating any course with more than 1000 valid combinations as a special case.

The two principal effects of these changes were an increase in the core required to store the course section data and a simplification of the scheduling algorithm for all courses but the special ones. Based on data from the last three terms, the increase in core needed was 9.8%, from an average of 15,152 to 16,649 words.

The third stage of development was undertaken to improve the efficiency and speed of the scheduling algorithm. Modifications made included adding a common restriction word and a common time vector for each course which were the "logical and" product of the restriction words and time vectors of the combinations of the course respectively. These modifications lowered the average number of attempts to schedule a student from approximately 300 during the stage two runs to 58 during the stage three runs.

II. SOURCE AND PREPARATION OF SCHEDULING DATA

Schedule of Classes Course Structure

The course data in the schedule of classes (SOC) file is the underlying structure upon which the scheduling algorithm is based. This file consists of course section data for the 120 departments on campus. Within these departments, there are approximately 1900 courses with 6300 sections being offered each term.

The SOC file is maintained in the following sort order: department, course, section, and type of section. The section and type of section numbers are assigned to distinguish seven basic course arrangements. These are:

1) a lecture only course,
2) a lecture and recitation course,
3) a lecture and lab course,
4) a lecture, recitation,and lab course,
5) a recitation and lab course,
6) a PE activity course, and
7) a military science course which requires a drill section.

There are eight type-of-section codes used to indicate these course arrangements and the type of sections within each of them. These are:

1 - lecture section of a lecture only course,

2 - lecture section of a lecture/lab course,

3 - lecture section of a lecture/recitation course,

4 - lecture section of a lecture/recitation/lab course,

5 - lab section,

6 - PE activity section,

7 - recitation section, and

8 - recitation section of a recitation/lab course.

The course arrangements are further complicated by the varying relationships between lecture and recitation and/or lab sections actually in use. To standardize these relationships for scheduling purposes a code, called an expansion code, is included in the data for each lecture section of a course. This code is used as follows:

2 - any lecture section can be scheduled with any recitation and/or lab section,

4 - lab and recitation sections are paired and any pair can be scheduled with any lecture,

6 - recitation, lab, or recitation-lab pairs are to be scheduled with specific lecture sections,

7 - a drill section is to be scheduled with the lecture section,

8 - a drill section is to be scheduled with the lecture section.

To handle reserve courses properly, as well as others similar to them in which different students may take the same course for different

credit hours, the following expansion code was added to the above:

1 - each section of a lecture only course has a different number of credit hours.

The valid combinations of section types and expansion codes allow the scheduling algorithm to handle 13 course arrangements. These are:

1) lecture only course in which all sections have the same credit hours,
2) lecture only course in which each section has different credit hours,
3) lecture/recitation course in which any lecture and recitation section may be scheduled,
4) lecture/recitation course in which recitations must be scheduled with specific lecture sections,
5) same as 3 for lecture/lab course,
6) same as 4 for lecture/lab course,
7) lecture/recitation/lab course in which any lecture, recitation, and lab sections may be scheduled,
8) lecture/recitation/lab course in which recitation-lab pairs may be scheduled with any lecture section,
9) lecture/recitation/lab course in which recitation-lab pairs must be scheduled with specific lecture sections,
10) PE activity course,

11) military science course for which a drill section must be scheduled,

12) same as 3 for recitation/lab course, and

13) same as 4 for recitation/lab course.

Course Data Preparation for Scheduler

To minimize the file and core requirements of the scheduling program, only the data needed to identify sections and the restrictions on those sections is used by the scheduling program. The following data elements are those so required:

1) department code (4 characters),

2) course number and alpha (4),

3) section number (3),

4) type of section (1),

5) credit hours (2),

6) expansion code (1),

7) time section offered converted to binary vector (24),

8) sex restriction (1),

9) class restrictions (7),

10) school restrictions (12), and

11) number of stations (3).

To take the required data elements and generate the data file actually used by the scheduling program, a program called SKEDFILE

was written. This program edits the data fields in the SOC file, checks for valid combinations of lecture, recitation, and/or lab sections, and generates the data file and a file containing data on canceled courses. The scheduling process cannot proceed until all errors identified by this program are corrected in the SOC file. As the sections are checked, any time conflicts between lecture and recitation and/or lab sections are identified. These conflicts may or may not indicate that one or more sections have the wrong time offered specified.

The course data file generated by this program contains the following arrays and data elements:

1) department codes,
2) pointer to the first course in the department,
3) course numbers,
4) pointer to the first word of section data for the course,
5) number of sections of the course,
6) common restrictions for the course,
7) pointer to the common time vector for the course,
8) number of words of section data for each school,
9) pointer to the first word of section data for each school,
10) section data for all courses,
11) time vector data, and
12) special data for military science drills.

A more detailed description of these arrays can be found in Appendix C.

The canceled course file contains the following elements for each such course:

1) department number,
2) course number,
3) pointer to section data for the course,
4) scheduling sequence number of the student who closed the course (zero for canceled courses), and
5) number of requests for the course after closing.

During each scheduling run, this file is updated with the latest data.

Section Data Generation

The department and course arrays are used to validate the course requests as they are processed by the scheduler and to point to the section data for the course, but the data used by the scheduling algorithm is the section data. The combination technique for storing this data was chosen for several reasons. First, it allows all courses to be processed in the same way, with, therefore, a less complex algorithm. One of the problems with the original version of the scheduling algorithm was that it did not keep track of the relationships between lecture and recitation or lab sections properly. Second, the scheduling algorithm can run faster because it does not have to

check which recitations and/or labs go with which lecture sections every time it tries to schedule a course. Finally, it simplifies the checking involved in scheduling those courses with labs or recitations which have only one lecture, lab, or recitation section.

The section data for a course is composed of five different word types, the number and order of which are determined by the type of course and the expansion code for that course. These words are:

1) Pointer word. This word contains a pointer to the restriction and time vector data for a specific combination of sections for the course. It also contains a percentage figure indicating the percent of stations left for the combination. There is one pointer for each combination and these pointers are sorted into increasing percentage order each time a student is scheduled into the course. This sorting is the technique used to maintain section balancing. The percentage is figured by one of the following formulas:

   a) for lecture only courses:

      % = [(stations scheduled)*100]/[original stations]

   b) for lecture/recitation and lecture/lab courses:

      % = [((lecture stations scheduled) + (recitation/lab stations scheduled))*100] /[(original lecture stations) + (original recitation/lab stations)]

c) for lecture/recitation/lab courses:

% = [((lecture stations scheduled) + (recitation stations scheduled) + (lab stations scheduled))*100] / [(original lecture stations) + (original recitation stations) + (original lab stations)]

2) Restriction word. This word has bits set if students meeting the associated criterion cannot take the section. This word appears in the section data once for each combination.

3) Time vector pointer word. This word points to a six word time vector in which bits are set for the half-hour periods during the week when the sections meet. It also contains the section number of the lecture section in the combination. This word appears in the section data once for each combination.

4) Station word. This word contains the original number of stations and the number of stations scheduled. This word appears in the section data once for each section of the course.

5) Section word. This word contains the section number and the ordinal number of a section. This word appears in the section data once for each section of a course having recitations and/or labs.

A more detailed description of the section data structure appears in Appendices B and C.

III. THE SCHEDULING PROCESS

General Scheduling Process

The general scheduling process is indicated by Figure 1. The first step is to read a control card which specifies the sequence number of the first and last students to be scheduled during the current run, the maximum number of attempts to make before giving a student an incomplete schedule, the number of students to schedule between course data dumps, and the number of students to schedule between conflict summaries.

Next, the current course data is read into core from a data file. This is the file initially generated by the SKEDFILE program and at least one updated set of course data is appended to it during each scheduling run. The last set of data on the file is assumed to be the current data for any scheduling run. The sequence number of the last student scheduled is written out with each set of data so this number must be one less than the first sequence number on the control card for the current run. If it is not, an error message is printed and the program stops.

After the course data is read into core, the program skips out on the student requests file to the student with the sequence number specified on the control card.

Then students are scheduled until the end of the requests file is

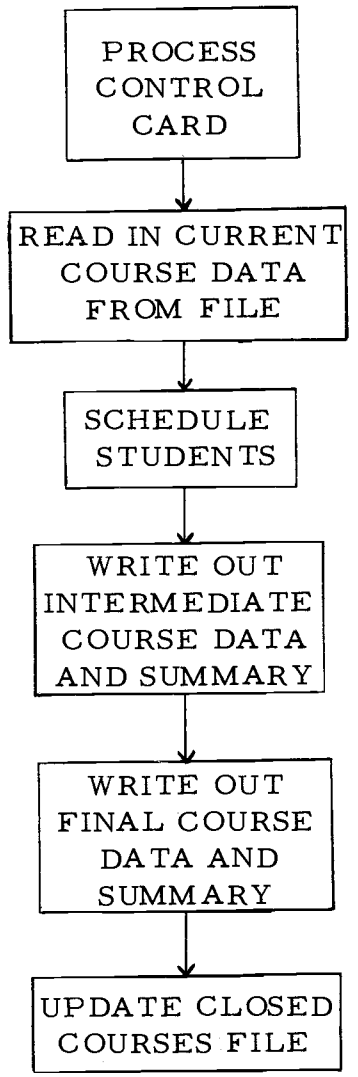


Figure 1. General scheduling process.

reached or the student specified by the second sequence number on the control card is scheduled.

At the breaks indicated by the control card either an intermediate conflict summary or an intermediate course data dump is generated. The course dump was initially included in the program to produce data for restarting the scheduling process in case of operating system or hardware failure. This option has not been used during the last five terms because there have been no such failures during scheduling and even if there were it is simpler now just to reschedule all students. With the entire scheduling run taking less than 30 minutes, time for this rerun would be available.

After the last student is scheduled, a final conflict summary is printed and a final course data dump is appended to the data file.

Finally, the closed courses file is updated with information on the courses closed during the current scheduling run and with information on requests processed for the other closed courses.

General Student Processing

If a student has no course requests, an error message is printed and the next student is processed. This processing follows the steps outlined in Figure 2. First the header data for the student is moved to the output array and then arrays and variables needed for scheduling are initialized. This header information contains data about

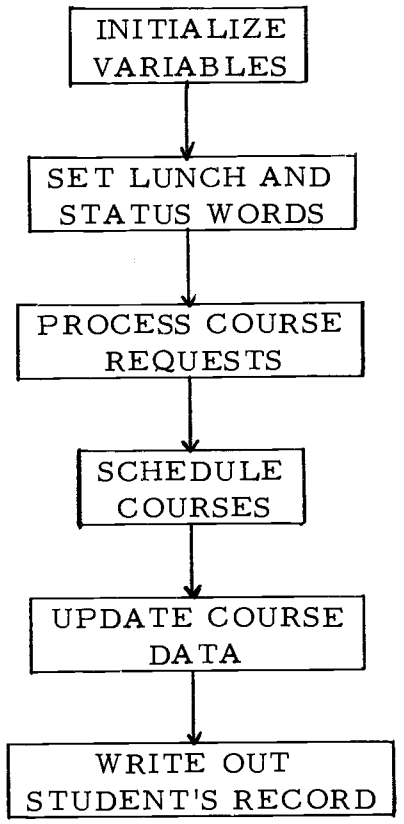


Figure 2. General student processing.

the student which is used by the billing program at a later stage of the registration process. A more detailed description of the requests file and the scheduled students file can be found in Appendix E.

After the scheduling arrays are initialized, a flag is set if the student has indicated he does not require a lunch hour. Having this flag set removes any time restrictions during the scheduling process other than free time request restrictions. The status word has bits set indicating the student's sex, school, and class. This word is generated for checking against the restriction words in the section data during scheduling.

The next step is to validate all course requests and to initialize the scheduling arrays with course data for all valid courses. This processing is described in more detail in the next section of this chapter.

Then any valid courses are scheduled by the algorithm described later in this chapter.

After the courses are scheduled, the course and section data for these courses is updated. This updating includes increasing the station counts for each section by one, recomputing the stations scheduled percentage for any combination of the course including those sections, resorting the combination pointers, and, finally, if the student closed the course, flagging the course as closed.

Finally, the conflict summaries are updated with the data for the student and the student's record is written out on file.

Initial Course Request Processing

The course requests are processed to validate them and to take care of special cases before the valid requests are scheduled. During this processing, if a student cannot take the requested course, the request data is moved to the output array with a code indicating the reason the course was not scheduled.

The processing follows the steps indicated in Figure 3, the first step being validating the department code in the course request against

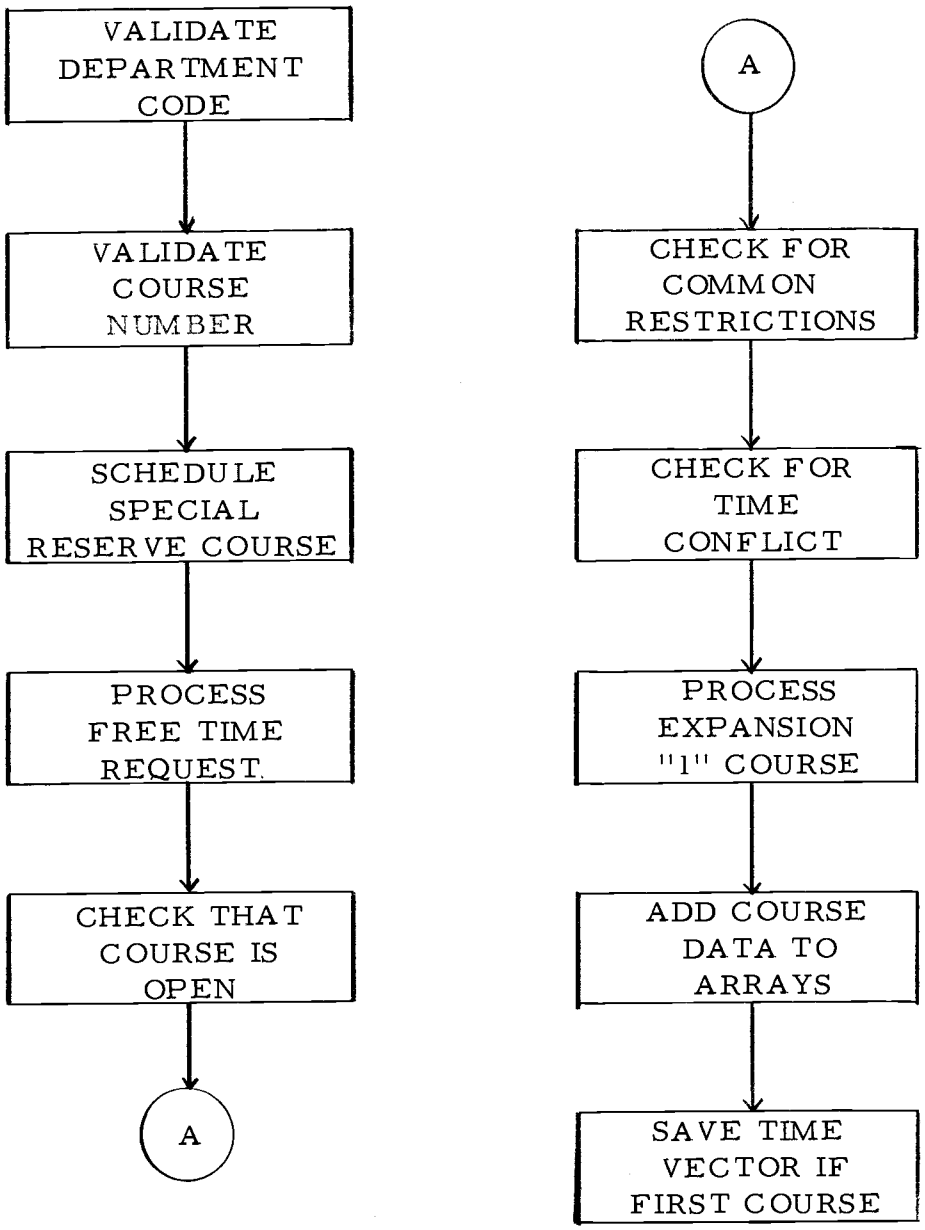


Figure 3. Initial course processing.

the department table read into core at the beginning of the scheduling run.

If the department code is valid, the course number is checked to make sure it is valid for the department. If the course is not found in the table of valid courses it is checked against a table of special reserve course numbers, i.e., 401, 403, 405, 406, 501, 503, 505, and 506. No course data is used for these courses, as they are assumed to be TBA (to be arranged) courses. If scheduling the course will not give the student more than 19 credit hours, the request is moved to the output array as a scheduled course. Otherwise the request is moved to the output array with the conflict code for excessive hours. No station totals are kept for these courses so there is no data to update. By handling these courses in this manner, over 2000 sections are effectively dropped from the SOC file, and at least 8000 fewer words of section data are needed in core during scheduling.

If the course request is for free time, the time vector for the course is "or"ed with any previous free time requests to generate the free time vector which will be used when scheduling the rest of the student's courses.

If the course is closed or the student cannot take the course because of a sex, class, or school restriction, the request is moved to the output array with the appropriate conflict flag.

If the common time vector for the course conflicts with the common time vector for the first course requested, the course cannot be scheduled. Because the priority of courses is established by their request order, the first course requested will always be scheduled if it was not rejected by one of the above tests.

If the course has a "1" expansion code, the course sections are searched for one having the credit hours indicated by the course request. If the course has no such section, or the section is closed, or the restrictions apply, the course is not scheduled. If the section is a TBA section, it is scheduled if it does not give the student more than 19 credit hours. If the section is not a TBA section, and it does not conflict with the common time vector for the first course, it is added to the scheduling arrays as a normal course.

If a course request survives the above checks and processing, it is a valid request and the data for the course is added to the scheduling arrays. These arrays include the following data for each course:

1) type and expansion codes,
2) number of combinations and sections,
3) number of credit hours,
4) pointers to the common time vector,
5) pointer to the combination currently scheduled,
6) pointer to the time vector for the combination currently scheduled,

7) pointers to the first and last combination pointers, and

8) pointer to the first combination tried during the current attempt to schedule the course.

If the first course request is being processed, the common time vector is saved for checking against the remaining requests.

No check is made during the scheduling process for duplicate course requests because there are courses for which students may legitimately have more than one request and in most other cases the second request would not be honored any way because of time conflicts.

Scheduling Algorithm

The basic approach taken by the scheduling algorithm is to find the first combination of each course which can be scheduled. When all combinations of previous courses have been tried and a course still cannot be scheduled, it is flagged as unschedulable and the next course is tried. The algorithm is as follows:

1) If the course is a PE activity course and one is already scheduled, flag this one and do not process again.

2) If scheduling the course would give the student more than 19 credit hours, flag the course and get the next request.

3) If the common time vector for the course conflicts with the common time vectors of the previously scheduled courses, flag the course as unschedulable and get the next request.

4) If the course has more than one combination, go to step 5. Otherwise, if the course can be scheduled without a time conflict, do so and get the next request. If the the course has a time conflict with the previously scheduled courses, and they cannot be scheduled in any other way, flag the course as unschedulable and process the next request. Otherwise, save the status of the currently scheduled courses and go to step 6.

5) Process each combination of the course as indicated below until either a combination is scheduled, or there are no more combinations to try. If a combination is scheduled, get the next request. If there are no more combinations to try, save the status of the currently scheduled courses and go to step 6. A combination is checked as follows:

   a) skip combination if closed,

   b) skip combination if restrictions apply,

   c) skip combination if it conflicts with free time requests,

   d) skip combination if it has a time conflict with previously scheduled courses,

   e) schedule combination and get next request.

6) At this point, backing up is necessary to search for

other ways of scheduling the courses requested before the one that could not be scheduled. This is done in the following way:

a) Unschedule each previous course until one is found which allows a combination of the current course to be scheduled. Scheduled courses with only one combination and courses flagged as unschedulable are left alone during this backup process. If a course is found, the next combination of that course is tried starting over at step 5.

b) If no such course is found, and no course so far processed has a combination which conflicted with the student's free time requests, flag the current course as unschedulable, restore the status of the course data to what it was before backing up, and get the next request.

c) Otherwise, remove the free time restriction from the first course which had combinations which conflicted with the free time requests, and start over at step 5 with this course.

To handle the military science courses which require a drill section to be scheduled and the special courses which would have more than 1000 valid combinations, there are two duplicates of steps 4

through 6 for scheduling the drill section and the recitation and lab sections respectively.

IV. EVALUATION OF THE SCHEDULING PROCESS

Evaluation of the Scheduling Algorithm

There are two principal areas in which the performance of the scheduling algorithm can be judged. These are wall clock time and section balancing. Other factors such as number of completed schedules are more a function of administration policies than the scheduling algorithm.

The wall clock time has definitely been within acceptable limits. It has dropped from approximately an hour and a half during the stage two runs to an average time of 22 minutes to schedule an average of 13, 900 students during the last three terms. This time was obtained running with an average of 15-20 other users concurrently under OS3.

The section balancing has also been within acceptable limits. For 90% of the sections, the number of students scheduled into sections of the same course are within two of each other. Most of the courses in which this level of balancing is not reached usually have one or more limiting factors. Some of these are:

1) Sections within the course are restricted to different groups of students.
2) One or more of the recitation or lab sections may be offered at the same time as one of the lecture sections.

3) One lecture section may have more recitation or lab sections that can be scheduled with it.

The principal drawback of the present algorithm is the lack of choice the student has over which section he will be scheduled for. This problem will be somewhat alleviated with the introduction of instructor preference cards by which the student can indicate the instructor he wants for a specific course. On the other hand, a student now has a better chance of getting a section in a course he wants because the sections are balanced, so there is more chance of there being a section open which is offered at a time he can take it.

To give some idea of the number of completed schedules each term, the average percentages for the 1971-1972 school year are as follows: 65% of the students were scheduled for as many hours as they requested, 68.8% were at most one hour short, 86.1% were at most three hours short, and 96.4% were at most six hours short.

V. PLANNED AND POSSIBLE FUTURE MODIFICATIONS

Instructor Preference Processing

The principal modification of the scheduling algorithm which is currently under development is the introduction of instructor preference cards. These cards will enable the student to specify which instructor he desires to have in any course with more than one instructor.

This feature was implemented by adding an instructor code to the course request data as generated by the packet processor, and adding the same codes to the combination pointer words in the section data for the courses with more than one instructor. A percentage field was also added to the scheduling control card. This percentage can be 0 to 100 and it specifies the percentage of stations scheduled in a course, above which an instructor preference for that course will not be specifically honored. Therefore, a zero in the control card indicates that no checking will be done during scheduling for instructor preferences by the students, and 100 in the control card indicates that an attempt will be made to honor all instructor preferences submitted by the students.

The data for the test run was collected during the Winter Term 1972 registration, in which 7638 valid instructor preference cards were submitted. Two test runs have been made, one with zero and

one with 100 percent specified, the results from which are as follows:

% of Requests honored	Average tries	WCT	CPU	Percentage
85 (6519 out of 7638)	101	33 min.	1231 sec.	100
36 (2742 out of 7638)	36	29 min.	735 sec.	0

The 36% in the second run are effectively those requests honored by chance. There was also a 1% increase in the total number of course requests which were not honored because of a conflict of some type. This increase is probably caused by the scheduling algorithm's not handling all cases properly, and work is still being done to correct this.

Another approach to instructor preference which could be implemented in addition to the current one, is for the students to submit cards indicating instructors they do not want to have for various personal reasons. This approach could be implemented without much difficulty and the scheduling time would not increase appreciably. The major decision which would have to be made if it were implemented is whether it would be an absolute restriction, meaning that the course would not be scheduled if the only instructor available was the one the student would rather not have, or a restriction which would be relaxed at some point in the scheduling process if necessary.

Course Alternates

A possible area of improvement in the scheduling process would be the introduction of course request alternates. These would allow the student to indicate alternate courses he wants scheduled if his primary requests cannot be honored. There are currently alternates for PE activity courses where only one of any number requested will be scheduled, but no other work has been done in this area other than identifying possible ways of implementing alternate requests. Some of these are:

1) The student would specify the maximum number of credit hours he wants to be scheduled for.
2) The student would submit alternates for specific courses.
3) The student would submit alternates for any course.

The implementation of these or any other methods would depend on the specific administration policies which governed their use. The scheduling run time would probably rise significantly if a majority of students submitted alternates.

Blue Sky Possibilities

There are many different approaches to the scheduling problem, but for most of them the advantages and disadvantages are hard to

assess. One possible approach would be on-line scheduling. This would allow the student to know exactly which courses and which sections he would be scheduled for without the need for adding courses because of other courses being closed. A test system for a one terminal operation has been written and it could be developed into a full system. The major disadvantage of this approach is the number of terminals which would be necessary to get all students scheduled in a reasonable amount of time. This could be partially solved by starting this registration process early in the preceding term.

Another area in which some system could be developed would be giving the student information during the scheduling process on which courses he should or should not be taking, based on courses he has previously taken and the courses he needs to take for his degree. This system would supplement or replace the use of advisors, but it is questionable that a system could be developed which would be as effective as personal advising. One drawback to this type of system is the cost of collecting the initial data necessary and another is that the student would be subjected to another set of rules more restrictive than those to which he is currently subjected.

BIBLIOGRAPHY

Busan, Vincent A. An algorithm for class scheduling with section preference. Communications of the ACM 10:567-569. 1967.

Faulkner, Martin. Computer sectioning and class scheduling. Datamation 11:35-37. June, 1965.

Macon, N. and E. E. Walker. A Monte Carlo algorithm for assigning students to classes. Communications of the ACM 9:339-340. 1966.

Manlove, Donald and David W. Beggs, III. Flexible scheduling using the IndiFlexS model. Bloomington, Indiana University Press, 1965. 191 p.

Murphy, Judith and R. Sutter. School scheduling by computer, the story of GASP. New York, Educational Facilities Laboratories, 1964. 46 p.

Wiley, Deane and Lloyd K. Bishop. The flexibly scheduled high school. West Nyack, Parker Publishing Company, 1968. 208 p.

Winters, W. K. A scheduling algorithm for a computer assisted registration system. Communications of the ACM 14:166-171. 1971.

APPENDICES

APPENDIX A

SCHEDULE OF CLASSES DATA FIELDS AND CODE DESCRIPTIONS

Table 1. Schedule of classes file data fields.

Field content	Number of data characters
Section type	1
Term section offered	4
Department code	4
Course number	4
Credit hours	2
Section number	3
Department alpha code	4
Expansion code	1
Lecture section designator	3
Sex restriction code	1
P/N grading code	1
Time section offered	24
Fall term stations	3
Winter term stations	3
Spring term stations	3
Course title	19
Building	3
Room number	4
Instructor	12
Instructor's Social Security number	9
Class restrictions	6
School restrictions	12

Table 2. Type of sections.

Type code	Section description
1	Lecture section of course with only lecture sections.
2	Lecture section of course with lecture and lab sections.
3	Lecture section of course with lecture and recitation sections.
4	Lecture section of course with lecture, lab, and recitation sections.
5	Lab section.
6	Physical education activity section.
7	Recitation section.
8	Recitation section of course with recitation and lab sections.

Table 3. Valid combinations of section types within a course.

Type of first section	Type of remaining sections
1	1
2	2 and 5
3	3 and 7
4	4, 7, and 5
5	Illegal except for drill sections
6	6
7	Illegal
8	8 and 5

Table 4. Course expansion codes.

Code	Sections' relationships
1	Each section of a lecture only course has credit hours different from other sections of the same course.
2	Any recitation and/or lab section may be scheduled with any lecture section.
4	Recitation and lab sections are scheduled in pairs and any pair may be scheduled with any lecture section.
6	Recitation, lab, or recitation/lab pairs must be scheduled with specific lecture sections.
7	Drill section must be scheduled with lecture section.
8	Drill section must be scheduled with lecture section.

Table 5. Section order for valid type and expansion code combinations.

Type code	Expansion code	Order of sections
1	1	Increasing credit hours.
2, 3, or 8	2	All lecture sections followed by all lab or recitation sections.
2, 3, or 8	6	Lecture section followed by lab or recitation sections which must be scheduled with that lecture section.
2	7 or 8	Lecture sections only in section number order. Drill sections have a special course number.
4	2	All lecture sections, followed by all recitation sections, followed by all lab sections.
4	4	All lecture sections, followed by all recitation and lab sections paired by section number.
4	6	Lecture section followed by recitation-lab pairs which must be scheduled with that lecture section.

APPENDIX B

ORDER OF SECTIONS IN SCHEDULING COMBINATIONS

Table 6. Examples of lecture, recitation, and/or lab sections in the combinations generated by the SKEDFILE program.

Course types 2, 3, and 8; expansion code 2: course type 4; expansion code 4. Course with two lecture and three recitation and/or lab sections.

Combination 1: lecture section 1, recitation and/or lab section 1.
Combination 2: lecture section 1, recitation and/or lab section 2.
Combination 3: lecture section 1, recitation and/or lab section 3.
Combination 4: lecture section 2, recitation and/or lab section 1.
Combination 5: lecture section 2, recitation and/or lab section 2.
Combination 6: lecture section 2, recitation and/or lab section 3.

Course types 2, 3, 4, and 8; expansion code 6. Course with two lecture and five recitation and/or lab sections. The first three recitation sections must be scheduled with the first lecture section and the last two recitation sections must be scheduled with the second lecture section.

Combination 1: lecture section 1, recitation and/or lab section 1.
Combination 2: lecture section 1, recitation and/or lab section 2.
Combination 3: lecture section 1, recitation and/or lab section 3.
Combination 4: lecture section 2, recitation and/or lab section 4.
Combination 5: lecture section 2, recitation and/or lab section 5.

Course type 4; expansion code 2. Course with two lecture, two recitation, and two lab sections.

Combination 1: lecture section 1, recitation section 1, lab section 1.
Combination 2: lecture section 1, recitation section 1, lab section 2.
Combination 3: lecture section 1, recitation section 2, lab section 1.
Combination 4: lecture section 1, recitation section 2, lab section 2.
Combination 5: lecture section 2, recitation section 1, lab section 1.
Combination 6: lecture section 2, recitation section 1, lab section 2.
Combination 7: lecture section 2, recitation section 2, lab section 1.
Combination 8: lecture section 2, recitation section 2, lab section 2.

APPENDIX C

SCHEDULING DATA FILE ARRAY, WORD, AND BIT DEFINITIONS

Table 7. Scheduling course data file structure.

EOF	R1 SECTION DATA FOR SCHOOL 0 (FREE TIME)	R2 SECTION DATA FOR SCHOOL 1 (AGRICULTURE)	R3 SECTION DATA FOR SCHOOL 2 (BUSINESS)
. . .	R13 SECTION DATA FOR SCHOOL 12 (PE)	R14 SECTION DATA FOR SCHOOL 14 (HONORS)	R15 SECTION DATA FOR SCHOOL 20 (STUDY ABROAD)

IDEPTCT	IDEPT	IDEPTPT	ISCOURL	R16 ISCOURP	IDRILLD	IDRILLP	IDRILLSC	IDRILLV
L1	L1	L1	L2	L2	L3	L3	L3	L3

R17 ICOUR	R18 ICOURPT	R19 ICOURSCT	R20 ICOURVEC	R21 ICOURSTR
L4	L5	L6	L7	L8

R22 IVEC(1-5700)	R23 IVEC(5701-11400)	R24 ISTUM
L9	L9	L10

EOF	....	R 1	R 2	R 3	R 4	R 5	R 6	R 7	R 8	R 9	R 10	R 11	R 12	R 13	R 14	R 15	R 16	R 17	R 18	R 19	R 20	R 21	R 22	R 23	R 24	EOF	EOF

Table 8. Scheduling data array, word, and bit definitions.

L1	IDEPTCT	VARIABLE	Number of entries in IDEPT and IDEPTPT arrays.
	IDEPT	ARRAY(140)	Department codes.
	IDEPTPT	ARRAY(140)	Pointer to the data for the first course in the department in ICOUR, ICOURPT, ICOURSCT, ICOURVEC, and ICOURSTR arrays.
L2	ISCOURL	ARRAY(22)	Length of section data records for each school.
	ISCOURP	ARRAY(22)	Pointer to first word of section data for each school during scheduling.
L3	IDRILLD	ARRAY(3)	Military science department codes.
	IDRILLP	ARRAY(3)	Word from ICOURPT array for military science drill courses.
	IDRILLSC	ARRAY(3)	Word from ICOURSCT array for military science drill courses.
	IDRILLV	ARRAY(3)	Word from ICOURVEC array for military science drill courses.

(Continued on next page)

Table 8. (Continued)

L4 ICOUR ARRAY(2000) Closed and cancelled bits, course number and alpha. Blank alpha is set to zero.

Field definitions for word are as follows:

CLSD	CNCL	FIRST DIGIT	SECOND DIGIT	THIRD DIGIT	ALPHA
1	2	3 6	7 12	13 18	19 24

L5 ICOURPT ARRAY(2000) Course type, expansion, and P/N codes; pointer to section data for course during scheduling.

Field definitions for word are as follows:

COURSE TYPE	EXPAND CODE	P	POINTER TO SECTION DATA
1 4	5 8	9	10 24

L6 ICOURSCT ARRAY(2000) Credit hours and number of sections.

Field definitions for word are as follows:

	CREDIT HOURS	NUMBER OF LABS	NUMBER OF REC/LABS	NUMBER OF LECTURES
1	2 6	7 12	13 18	19 24

L7 ICOURVEC ARRAY(2000) Pointer to time vector that is common to all combinations of the course.

(Continued on next page)

Table 8. (Continued)

L8 ICOURSTR ARRAY(2000) Sex, school, and class restrictions common to all combinations of the course.

Field definitions for word are as follows:

SEX		SCHOOL														CLASS							
F	M	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	2	3	4	5	6	7	8
											0	1	2	3	4								
1	2	3													16	17							24

L9 IVEC ARRAY(6 900) Six word binary time vectors. Bit definitions for words are as follows:

M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
						1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	7	8	8	9	9	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8
0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1											WORD 1												24

M	M	M	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
1	1	2							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	9	0	7	7	8	8	9	9	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7
0	3	0	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1											WORD 2												24

U	U	U	U	U	U	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W
1	1	1	1	1	2							1	1	1	1	1	1	1	1	1	1	1	1
7	8	8	9	9	0	7	7	8	8	9	9	0	0	1	1	2	2	3	3	4	4	5	5
3	0	3	0	3	0	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1											WORD 3												24

(Continued on next page)

Table 8. (Continued)

W	W	W	W	W	W	W	W	W	H	H	H	H	H	H	H	H	H	H	H	H	H	H
1	1	1	1	1	1	1	1	2							1	1	1	1	1	1	1	1	1
6	6	7	7	8	8	9	9	0	7	7	8	8	9	9	0	0	1	1	2	2	3	3	4
0	3	0	3	0	3	0	3	0	0	3	0	3	0	3	0	3	0	3	0	3	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1												WORD 4											24

H	H	H	H	H	H	H	H	H	H	H	H	F	F	F	F	F	F	F	F	F	F	F	F
1	1	1	1	1	1	1	1	1	1	1	2							1	1	1	1	1	1
4	5	5	6	6	7	7	8	8	9	9	0	7	7	8	8	9	9	0	0	1	1	2	2
3	0	3	0	3	0	3	0	3	0	3	0	0	3	0	3	0	3	0	3	0	3	0	3
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1												WORD 5											24

F	F	F	F	F	F	F	F	F	F	F	F	F	F	F	S	S	S	S	S	S	S	S	S
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2					1	1	1	1	1
3	3	4	4	5	5	6	6	7	7	8	8	9	9	0	8	8	9	9	0	0	1	1	2
0	3	0	3	0	3	0	3	0	3	0	3	0	3	0	0	3	0	3	0	3	0	3	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1												WORD 6											24

L10	ISTUM	VARIABLE	Sequence number of last student scheduled.

Table 9. Word and bit definitions for course section data.

L11 Percentage of combination stations scheduled and pointer to restriction and time vector words for combination.

PERCENTAGE OF STATIONS SCHEDULED		COMBINATION POINTER
1 8	9 14	15 24

L12 Restriction word for combination.

SEX	SCHOOL	CLASS
F M	1 2 3 4 5 6 7 8 9 1 1 1 1 1 0 1 2 3 4	1 2 3 4 5 6 7 8
1 2	3 16	17 24

L13 Section number of lecture section in combination and pointer to time vector for combination.

LECTURE SECTION SECTION NUMBER	POINTER TO TIME VECTOR (IVEC)
1 10	11 24

L14 Original stations and stations scheduled for a section.

ORIGINAL STATIONS	STATIONS SCHEDULED
1 12	13 24

L15 Section number and ordinal number for a section.

ORDINAL NUMBER	SECTION NUMBER
1 12	13 24

L16 Pointer to lecture restriction and time vector data for courses with an expansion code of 1.

PERCENTAGE OF STATIONS SCHEDULED	ORDINAL NUMBER	ONE (1)
1 8	9 14	15 24

Table 9. (Continued)

L17 Section word for lecture sections with an expansion code of 6.

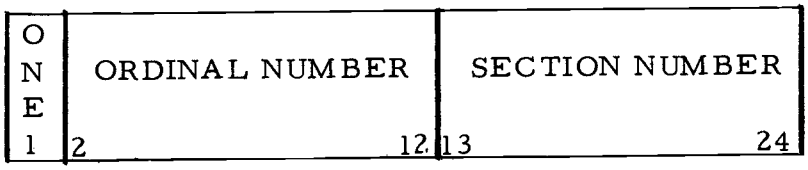


Table 10. Word arrangement of section data for various course types.

L18 Course types 1, 5, and 6 and type 2 with expansion code 7 or 8.

Arrangement for a course with two lecture sections would be:

POINTER L 1	POINTER L 2	RESTRICTION L 1	TIME VECTOR L 1
L11	L11	L12	L13

RESTRICTION L 2	TIME VECTOR L 2	STATION L 1	STATION L 2
L12	L13	L14	L14

L19 Course type 1 with expansion code 1.

Arrangement for a course with two lecture sections would be:

POINTER L 1	RESTRICTION L 1	TIME VECTOR L 1	STATION L 1
L16	L12	L13	L14

POINTER L 2	RESTRICTION L 2	TIME VECTOR L 2	STATION L 2
L16	L12	L13	L14

L20 Course types 2, 3, and 8 with expansion code 2 and course type 4 with expansion code 4.

Arrangement for a course with one lecture and one recitation or one lab or one recitation/lab pair (one combination).

POINTER C 1	RESTRICTION C 1	TIME VECTOR C 1	STATION L 1	STATION L 1	STATION R/L 1
L11	L12	L13	L14	L15	L14

SECTION R/L 1
L15

Table 10. (Continued)

Arrangement for a course with two lecture sections and three recitations or labs or recitation/lab pairs (six combinations).

POINTER C 1 L11	POINTER C 2 L11	POINTER C 3 L11	POINTER C 4 L11	POINTER C 5 L11	POINTER C 6 L11
RESTRICTION C 1 L12	TIME VECTOR C 1 L13	RESTRICTION C 2 L12	TIME VECTOR C 2 L13	RESTRICTION C 3 L12	TIME VECTOR C 3 L13
RESTRICTION C 4 L12	TIME VECTOR C 4 L13	RESTRICTION C 5 L12	TIME VECTOR C 5 L13	RESTRICTION C 6 L12	TIME VECTOR C 6 L13
STATION L 1 L14	SECTION L 1 L15	STATION L 2 L14	SECTION L 2 L15	STATION R/L 1 L14	SECTION R/L 1 L15
STATION R/L 2 L14	SECTION R/L 2 L15	STATION R/L 3 L14	SECTION R/L 3 L15		

L21 Course types 2, 3, and 8 with expansion code 6 and course type 4 with expansion code 6.

Arrangement for a course with one lecture and one recitation, lab, or recitation/lab pair (one combination).

POINTER C 1 L11	RESTRICTION C 1 L12	TIME VECTOR C 1 L13	STATION L 1 L14	SECTION L 1 L17	STATION R/L 1 L14
SECTION R/L 1 L15					

Table 10. (Continued)

Arrangement for a course with two lectures and three recitations, or labs, or recitation/lab pairs (three combinations).

POINTER C 1	POINTER C 2	POINTER C 3	RESTRICTION C 1	TIME VECTOR C 1	RESTRICTION C 2
L11	L11	L11	L12	L13	L12

TIME VECTOR C 2	RESTRICTION C 3	TIME VECTOR C 3	STATION L 1	SECTION L 1	STATION R/L 1
L13	L12	L13	L14	L17	L14

SECTION R/L 1	STATION R/L 2	SECTION R/L 2	STATION L 2	SECTION L 2	STATION R/L 3
L15	L14	L15	L14	L17	L14

SECTION R/L 3
L15

L22 Course type 4 with expansion code 2 (less than 1000 combinations).

Arrangement for a course with one lecture, one lab, and one recitation (one combination).

POINTER C 1	RESTRICTION C 1	TIME VECTOR C 1	STATION L 1	SECTION L 1	STATION R 1
L11	L12	L13	L14	L15	L14

SECTION R 1	STATION LB 1	SECTION LB 1
L15	L14	L15

Arrangement for a course with two lectures, two labs, and two recitations (eight combinations).

POINTER C 1	POINTER C 2	POINTER C 3	POINTER C 4	POINTER C 5	POINTER C 6
L11	L11	L11	L11	L11	L11

Table 10. (Continued)

POINTER C 7 L11	POINTER C 8 L11	RESTRICTION C 1 L12	TIME VECTOR C 1 L13	RESTRICTION C 2 L12	TIME VECTOR C 2 L13
RESTRICTION C 3 L12	TIME VECTOR C 3 L13	RESTRICTION C 4 L12	TIME VECTOR C 4 L13	RESTRICTION C 5 L12	TIME VECTOR C 5 L13
RESTRICTION C 6 L12	TIME VECTOR C 6 L13	RESTRICTION C 7 L12	TIME VECTOR C 7 L13	RESTRICTION C 8 L12	TIME VECTOR C 8 L13
STATION L 1 L14	SECTION L 1 L15	STATION L 2 L14	SECTION L 2 L15	STATION R 1 L14	SECTION R 1 L15
STATION R 2 L14	SECTION R 2 L15	STATION LB 1 L14	SECTION LB 1 L15	STATION LB 2 L14	SECTION LB 2 L15

L23 Course type 4 with expansion code 2 (more than 1000 combinations).

Arrangement for a course with two lectures, two labs, and two recitations. A true course of this type would have something like 25 lectures, 4 recitations, and 35 labs.

POINTER L 1 L11	POINTER L 2 L11	RESTRICTION L 1 L12	TIME VECTOR L 1 L13	RESTRICTION L 2 L12	TIME VECTOR L 2 L13
STATION L 1 L14	STATION L 2 L14	POINTER R 1 L11	POINTER R 2 L11	RESTRICTION R 1 L12	TIME VECTOR R 1 L13
RESTRICTION R 2 L12	TIME VECTOR R 2 L13	STATION R 1 L14	STATION R 2 L14	POINTER LB 1 L11	POINTER LB 1 L11

Table 10. (Continued)

RESTRICTION LB 1 L12	TIME VECTOR LB 1 L13	RESTRICTION LB 2 L12	TIME VECTOR LB 2 L13	STATION LB 1 L14	STATION LB 2 L14

APPENDIX D

SCHEDULING RUN FLOWCHART AND CORE LAYOUT

Table 11. Flowchart for a scheduling run.

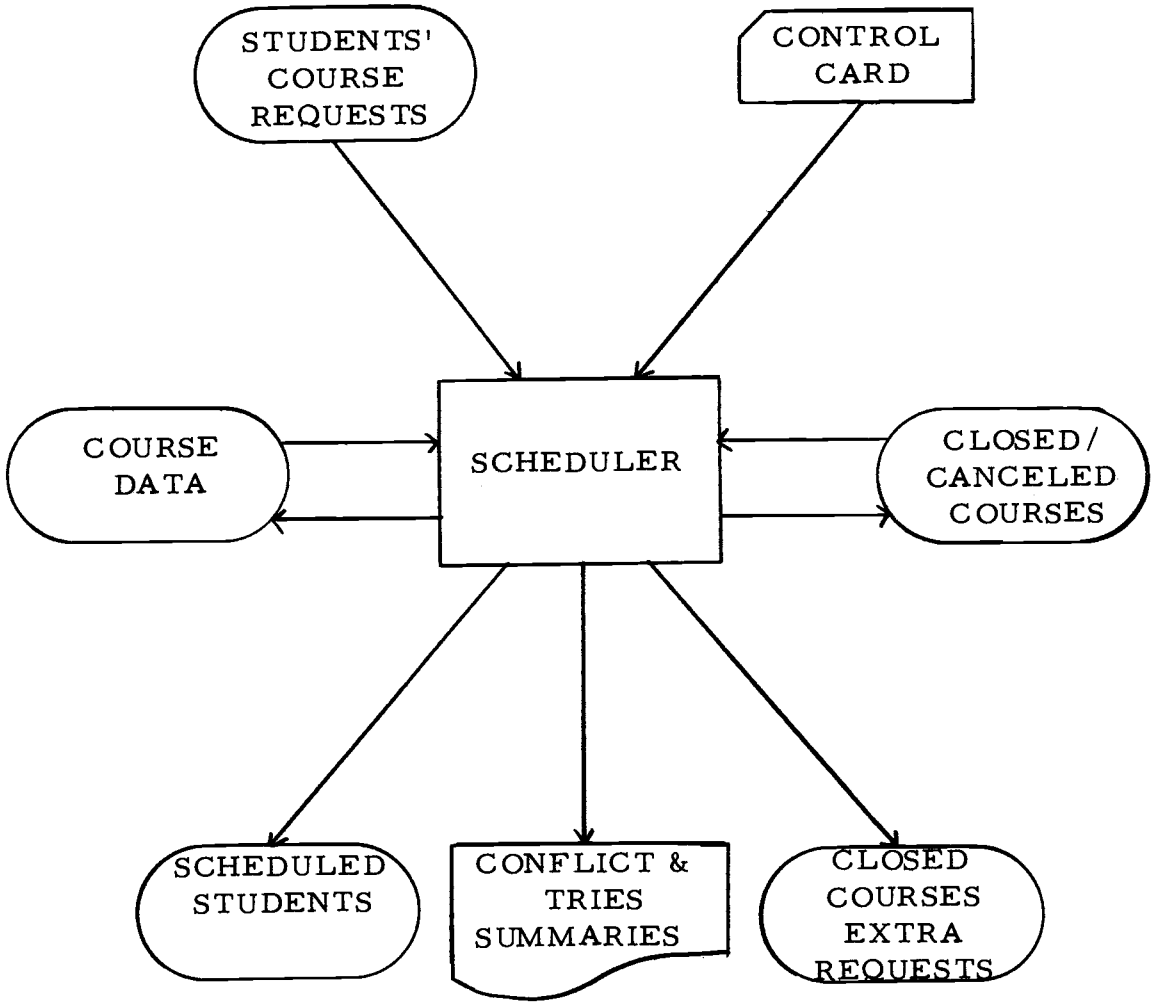
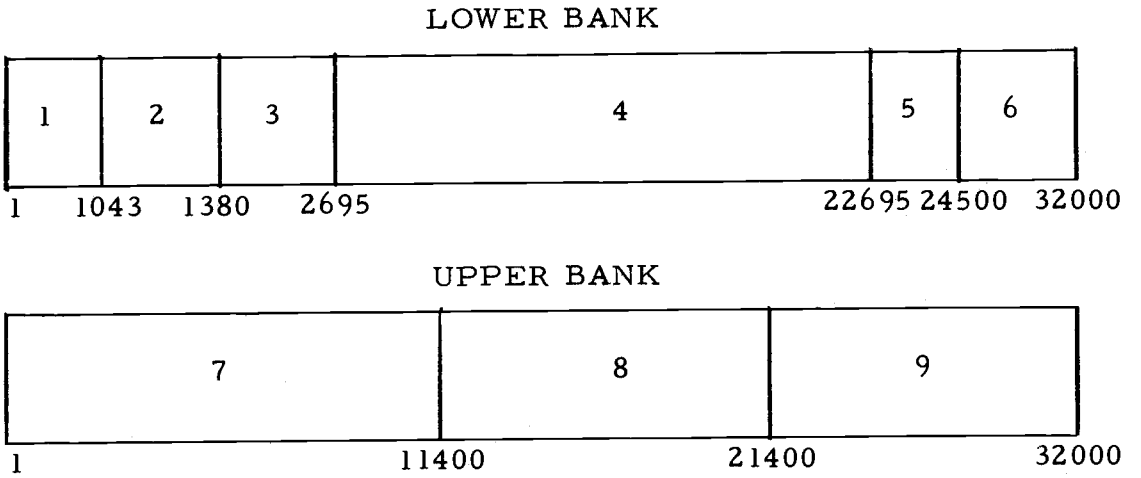


Table 12. Core layout during scheduling.



1. Arrays and variables used during scheduling of each student.
2. Department, school, and military science arrays.
3. Student course request file input array and scheduled student output array.
4. Section data.
5. Unused.
6. Program and subroutines.
7. Time vector arrays.
8. Course data arrays.
9. Unused.

APPENDIX E

STUDENT REQUESTS FILE AND SCHEDULED STUDENTS FILE RECORD LAYOUTS

Table 13. Student request file record layout.

Field content	Characters	Words
Social security number	1-9	1-3
Name	11-40	3-10
School	41	11
Class	42	11
Sex/marital status	43	11
Previous registration status	44	11
Residency code	45	12
Staff/Graduate assistant code	46	12
Concurrent enrollment/Auditor code	47	12
Foreign year fee code	48	12
Major department code	50-52	13
Service request word		14
Number of course requests		15
Scheduling input sequence number		16
Course request department code		17
Course request course number		18
Course request credit hours and S/U flag		19
Course request instructor/section specification		20

The last four fields are repeated once for each additional course request.

Table 14. Scheduled students file record layout.

Field content	Words
First 12 fields are same as specified in Table 13	1-14
Number of sections either scheduled or unscheduled	15
Scheduling input sequence number	16
Number of credit hours for which student is scheduled	17
Scheduling sequence number	18
Number of tries made to schedule student	19
Course section department code	20
Course section course number	21
Course section credit hours, S/U flag, and conflict flag	22
Course section section ordinal and section number	23

The last four fields are repeated once for each additional section the student was scheduled for or was not scheduled for because of some conflict.

APPENDIX F

SAMPLE CONFLICT SUMMARY REPORT AND CONFLICT DEFINITIONS

Table 15. Sample scheduling conflict and tries summary.

Type of conflict	Number of conflicts per student 1	2	3	4	5	Total students	Total conflicts
NTOF	27	1	0	0	0	28	29
CANL	393	14	1	0	0	408	424
CLOS	2344	710	189	45	7	3295	4546
ATMP	2	0	0	0	0	2	2
EXCS	172	12	4	0	1	189	213
SEXR	44	9	2	0	0	55	68
KLAS	776	87	11	2	0	876	991
SKOL	13	0	0	0	0	13	13
LHRQ	92	3	0	0	0	95	98
UCFL	1375	115	14	1	0	1503	1649
PELM	1076	641	4	0	0	1721	2370
FTRQ	173	51	7	3	0	234	308
SATU	1976	178	18	8	1	2181	2423
P/N	2289	335	32	1	0	2657	3059
TOTL	3786	1323	379	87	23	5598	8032

Tries	Count	Tries	Count
0- 499	13218	0- 49	12631
500- 999	30	50- 99	313
1000-1499	7	100-149	101
1500-1999	6	150-199	61
2000-2499	6	200-249	38
2500-2999	4	250-299	21
3000-3499	2	300-349	22
3500-3999	1	350-399	15
4000-4499	1	400-449	9
4500-4999	1	450-499	7
5000-LIMIT	8		

Students 1 to 13284 were scheduled in 576871 tries averaging 43 tries per student.

Table 16. Course conflict code definitions.

CANL	Course is in schedule of classes but all sections were closed before scheduling started.
ATMP	More than 100, 000 tries were made to schedule this course. Course is not scheduled.
EXCS	Course was not scheduled because scheduling it would give student more than 19 hours.
CLOS	All sections of the course were filled during prior scheduling.
LHRQ	No section of the course could be scheduled and at least one would not have left the student with a half hour for lunch if it had been scheduled.
NTOF	No such course is in the schedule of classes.
FTRQ	No section of the course could be scheduled which did not conflict with the student's free time requests. The indicated section was scheduled even though it conflicts with the free time requests.
UCLF	Course was not scheduled because available sections conflicted with sections of courses already scheduled for the student.
SATU	Scheduled course is to be graded on a satisfactory/unsatisfactory basis.
P/N	Scheduled course is to be graded on a pass/no pass basis.
SEXR	All open sections caused a conflict and at least one was limited to students of the opposite sex.
KLAS	All open sections caused a conflict and at least one was limited to students in some other specific class levels.
SKOL	All open sections caused a conflict and at least one was limited to students in some other specific schools.
PELM	Course was not scheduled because another PE activity course was already scheduled.
TOTL	Summary of students who had one or more of the first eight types of conflicts.

APPENDIX G

SUMMARY TOTALS FROM 1971-1972 SCHOOL YEAR SCHEDULING RUNS

Table 17. Summary totals from 1971-1972 school year scheduling runs.

	Fall	Winter	Spring
Number of courses offered	1683	1765	1845
Number of sections	3782	3880	3703
Number of words of section data	17490	16586	15871
Number of binary time vectors	1750	1425	1373
Number of words for binary time vectors	10500	8550	8238
Number of BCD time vectors	626	658	670
Number of instructors specified on file	686	728	770
Number of students scheduled	14828	13585	13284
Wall clock time for scheduling run	23 min.	25 min.	18 min.
CPU time for scheduling run	963 sec.	640 sec.	600 sec.
Average number of tries to schedule	67	45	43
% of students with complete schedules	65.8	67.6	61.7
% of students 1 hour or less deficient	70.5	70.5	65.4
% of students 3 hours or less deficient	85.3	88.0	85.0
% of students 6 hours or less deficient	95.9	97.5	95.7